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

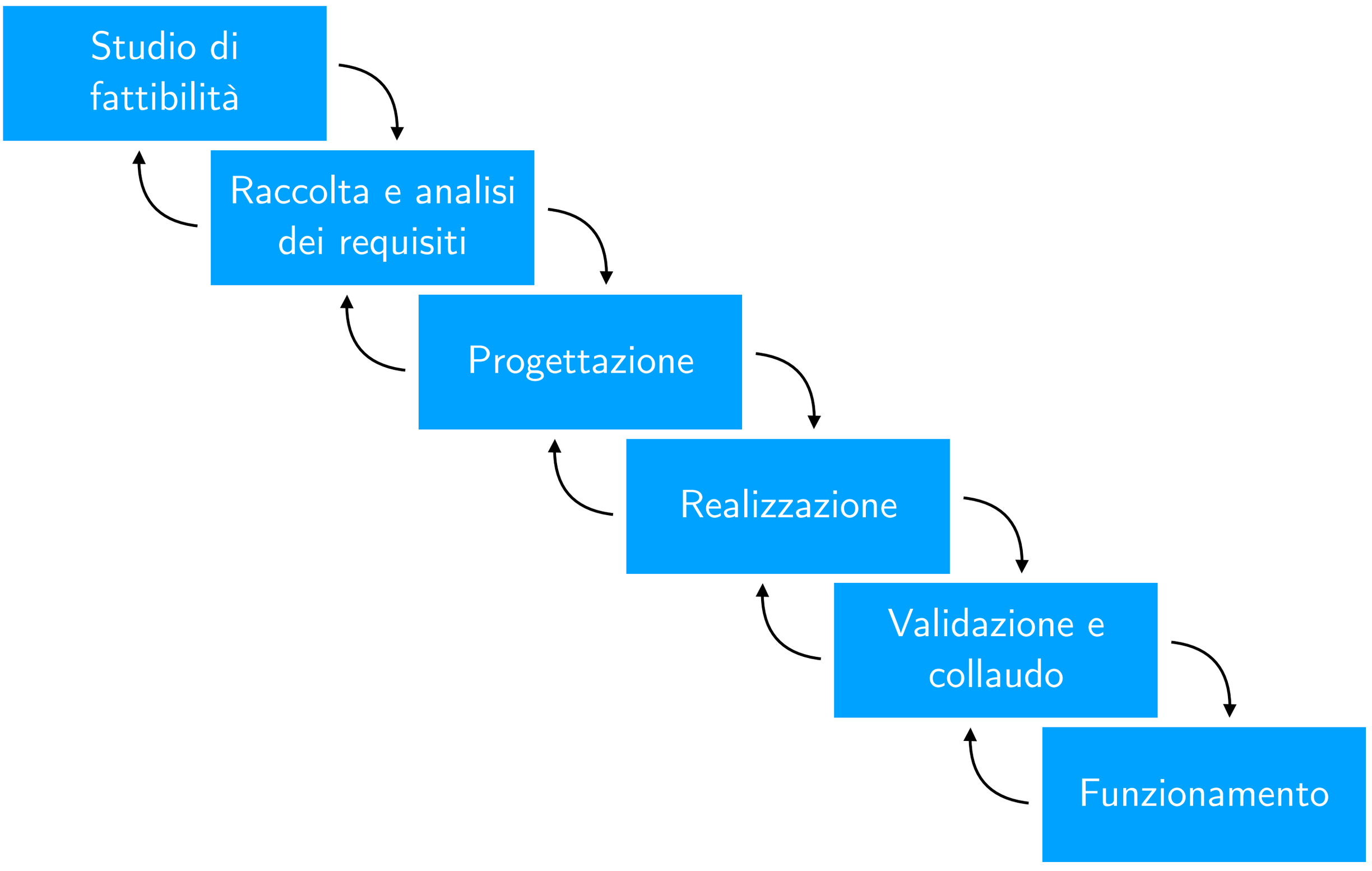
Un buon progetto

- 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** (generali 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

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.

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.

Frasi 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.

Frasi 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.

Frasi 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.

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.

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 **descriverne 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

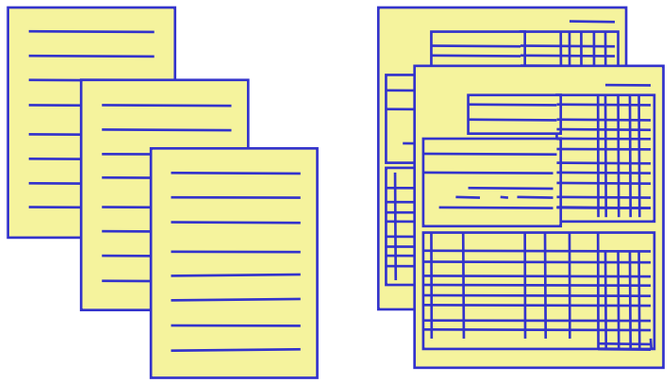
- In ogni **base di dati** esistono:
  - lo **schema**, sostanzialmente invariante nel tempo, che ne descrive la struttura
    - nel modello relazionale, le **intestazioni** delle tabelle
  - l'**istanza**, i valori attuali, che possono cambiare anche molto rapidamente
    - nel modello relazionale,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

Principali Tipi di Modelli

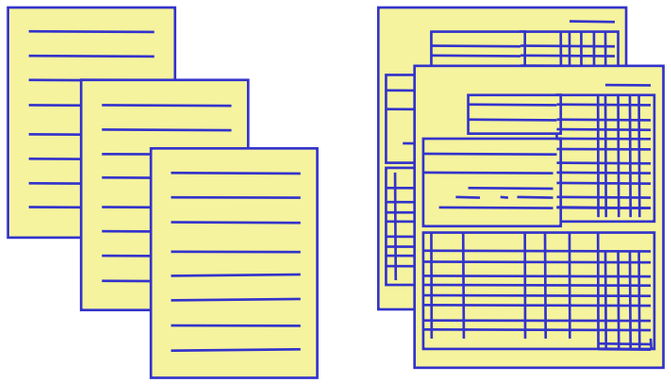
- **Modelli logici**: utilizzati nei DBMS esistenti per l'organizzazione dei dati
  - utilizzati dai programmi
  - indipendenti dalle strutture fisiche
  - esempi: **relazionale**, reticolare, gerarchico, a oggetti
- **Modelli concettuali**: permettono di rappresentare i dati in modo indipendente da 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

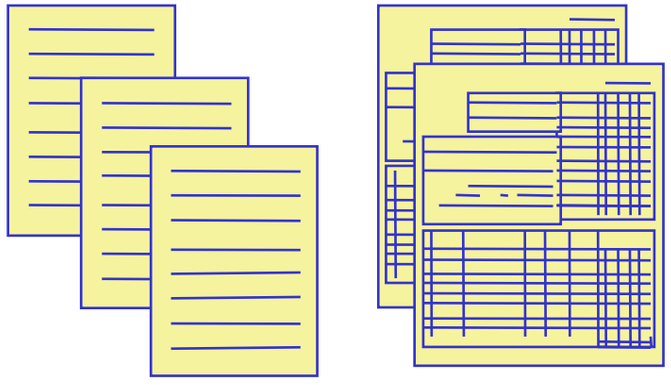


Passaggi tra Modelli

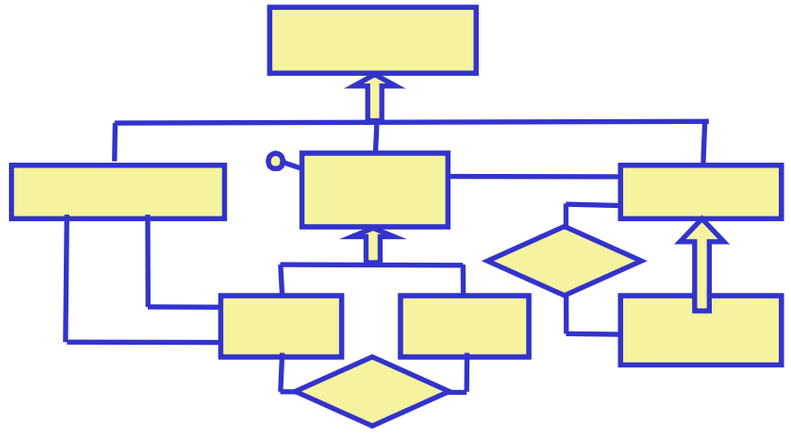


Progettazione concettuale

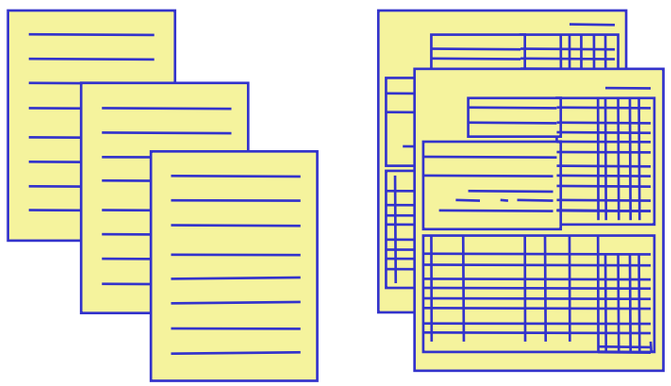
Passaggi tra Modelli



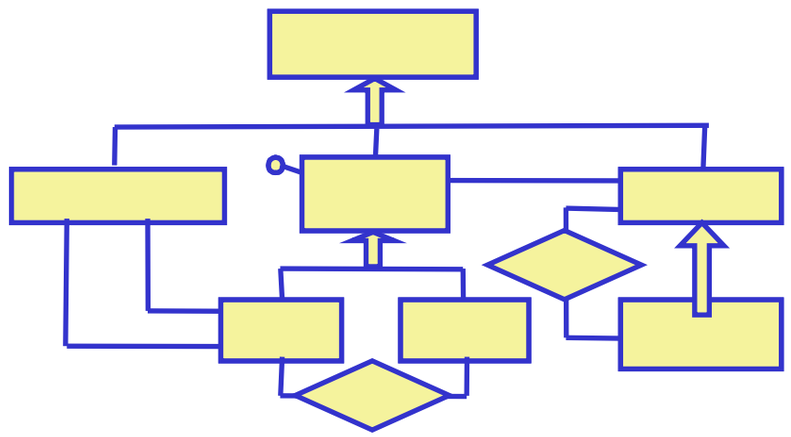
Progettazione concettuale



Passaggi tra Modelli

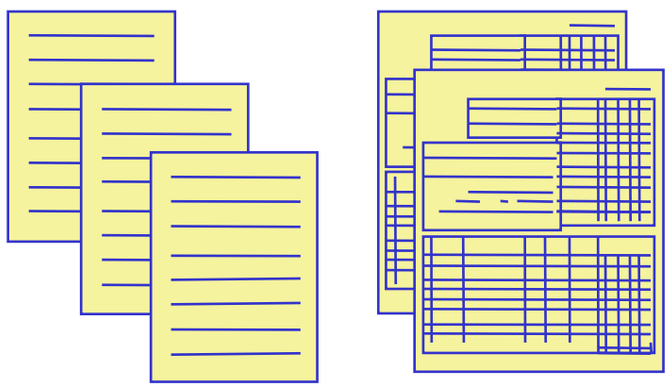


Progettazione concettuale

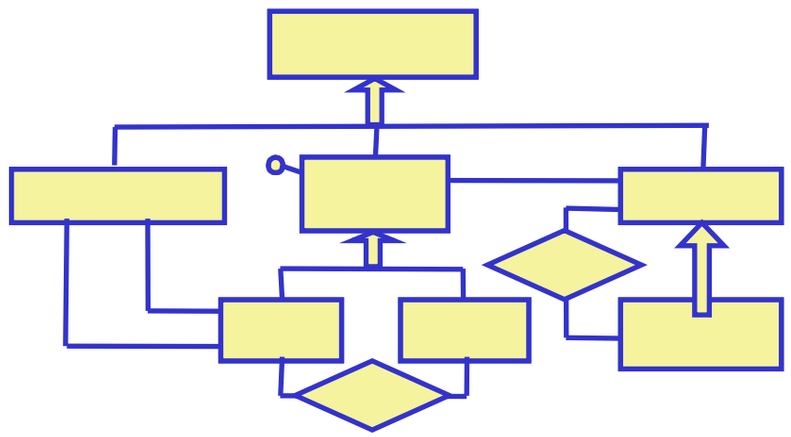


Progettazione logica

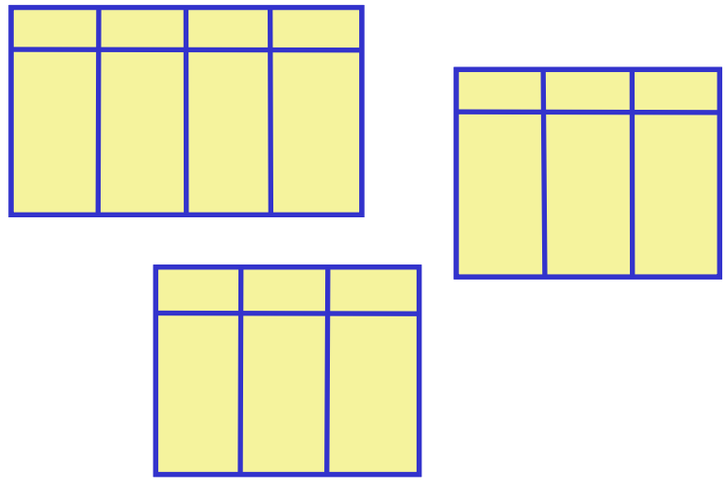
Passaggi tra Modelli



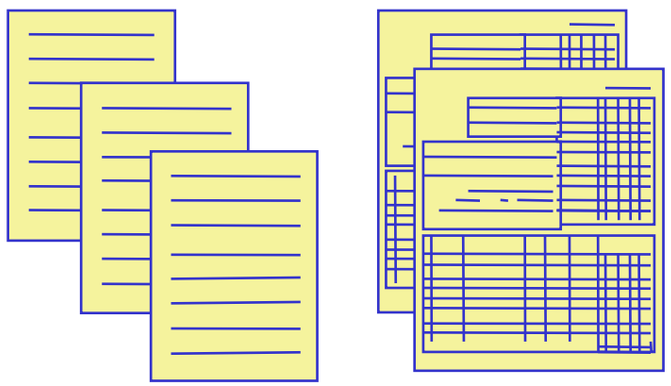
Progettazione concettuale



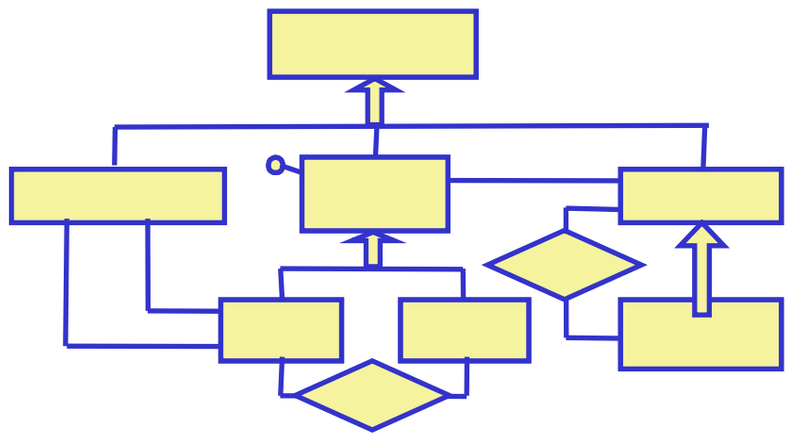
Progettazione logica



Passaggi tra Modelli

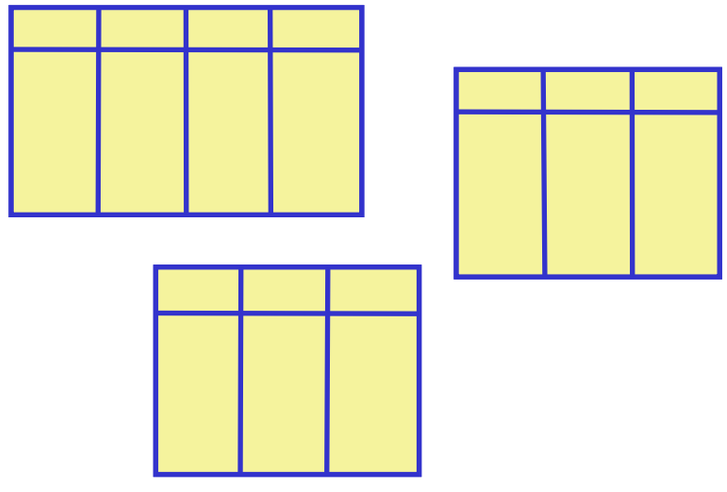


Progettazione concettuale

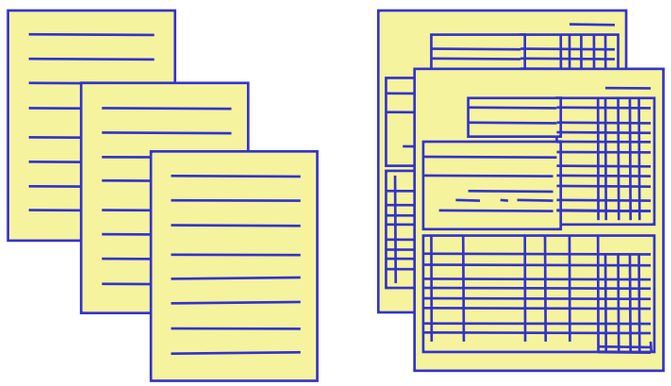


Progettazione logica

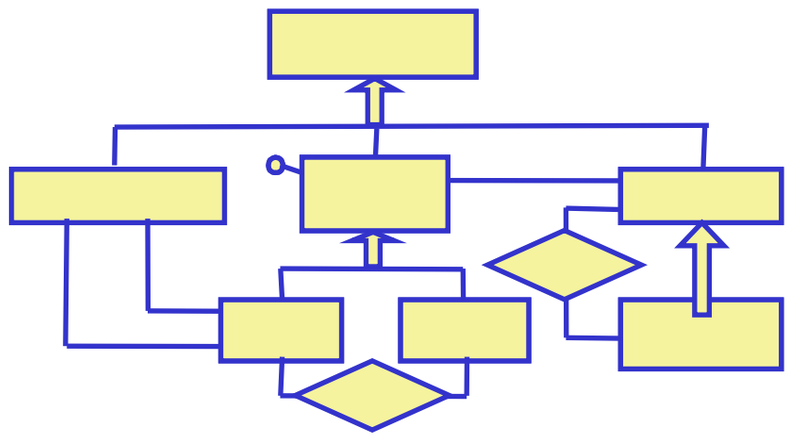
Progettazione fisica



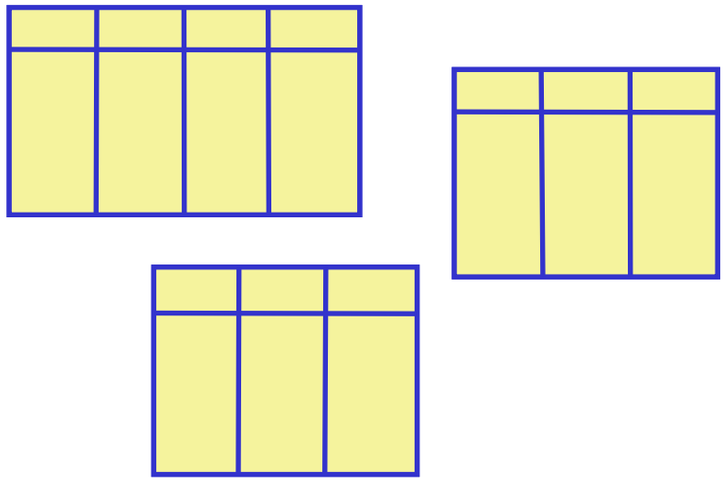
Passaggi tra Modelli



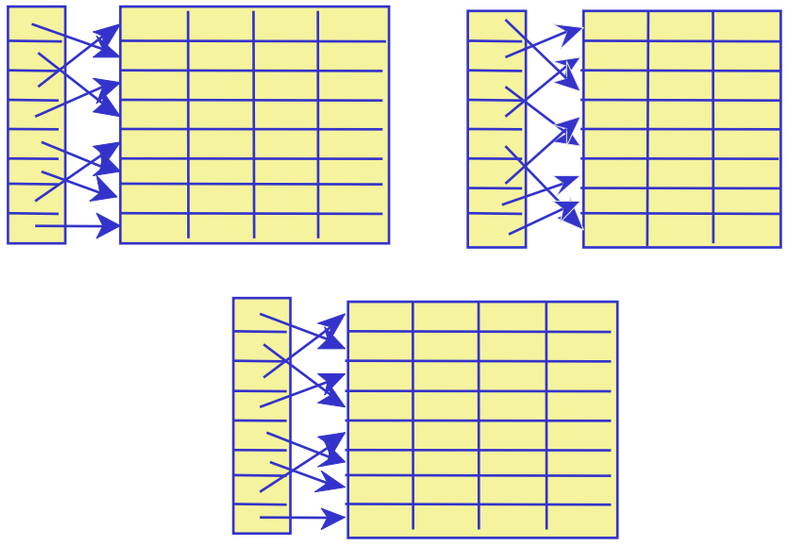
Progettazione concettuale



Progettazione logica



Progettazione fisica



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*

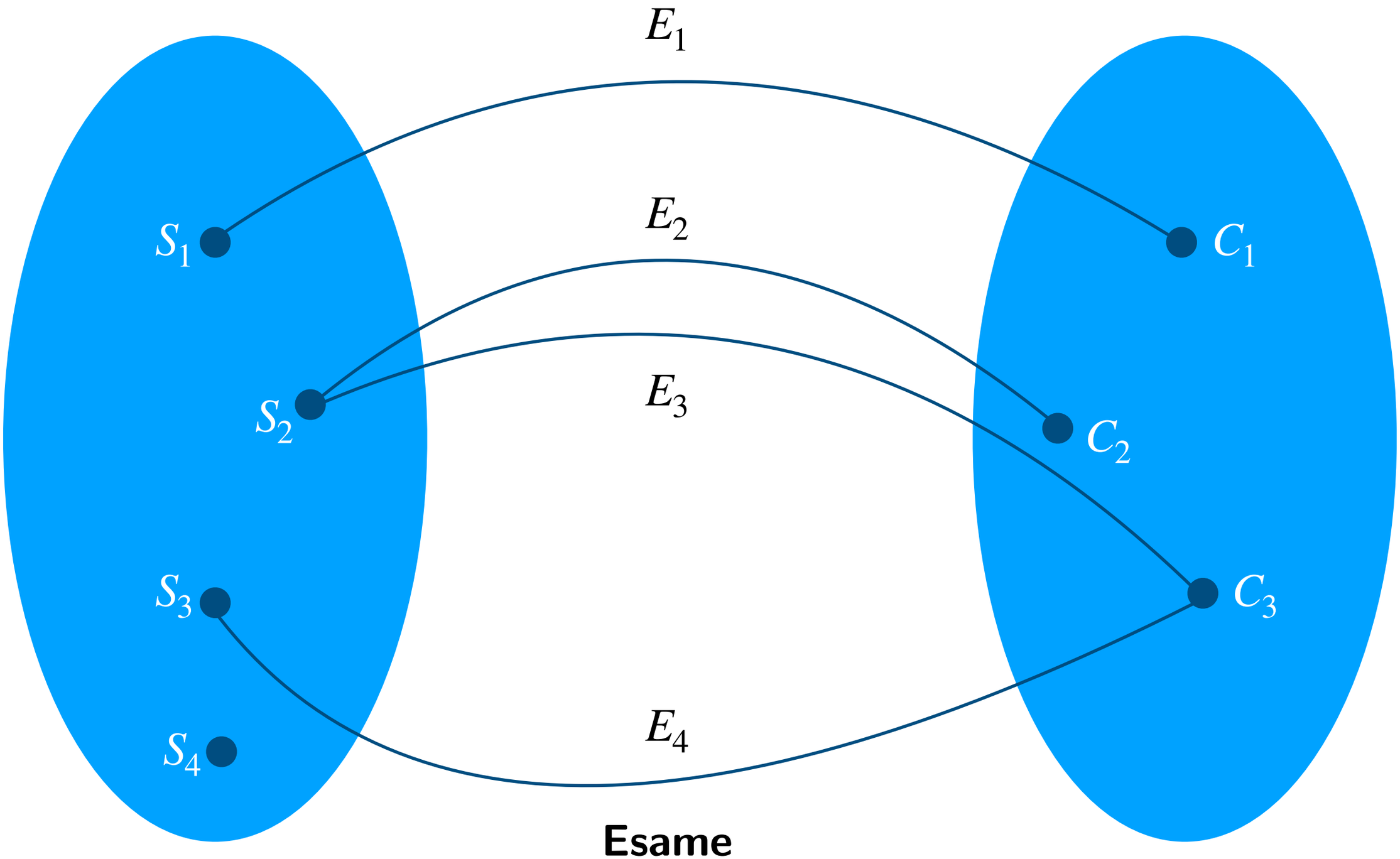
Caratteristiche delle *relationship*

- 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



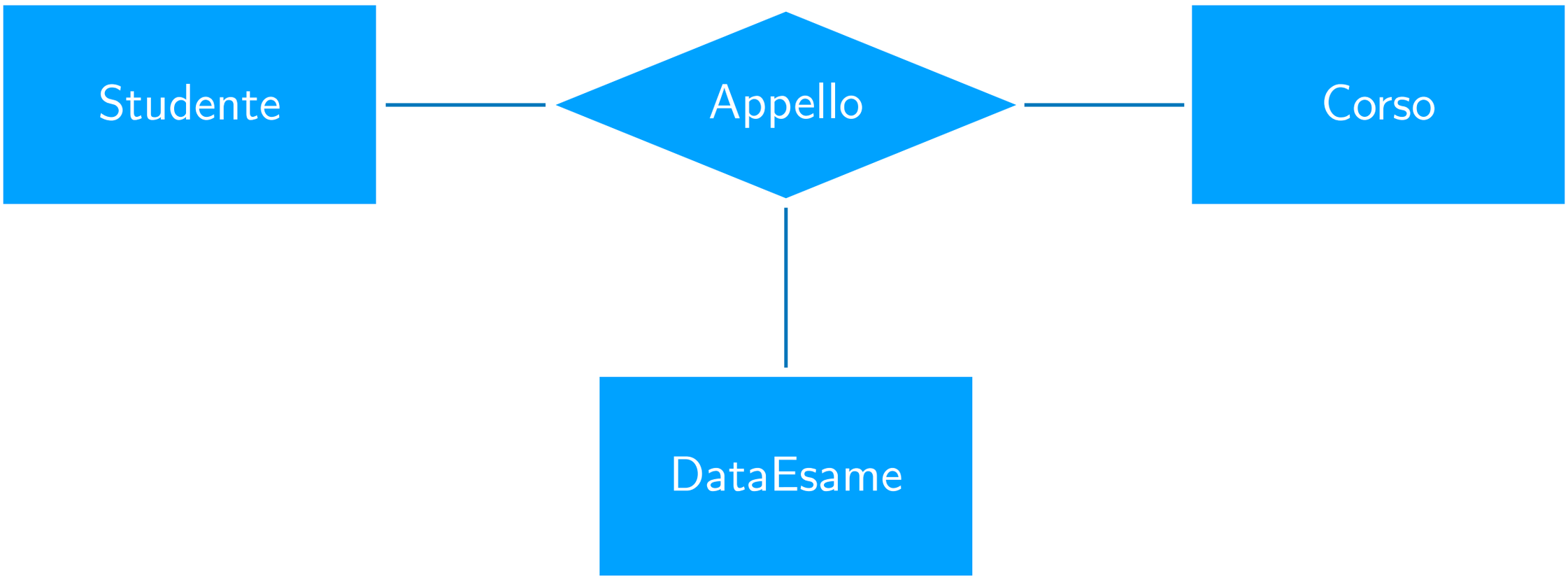
Esempio

- 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**

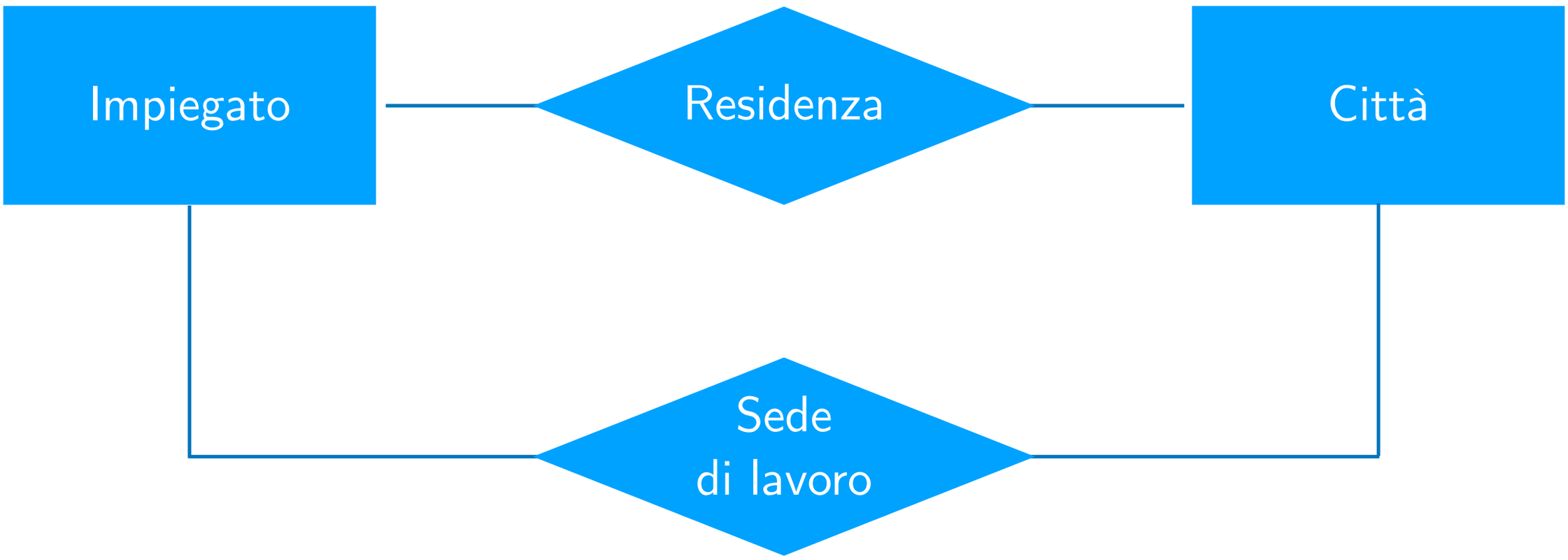
Prima Rappresentazione

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

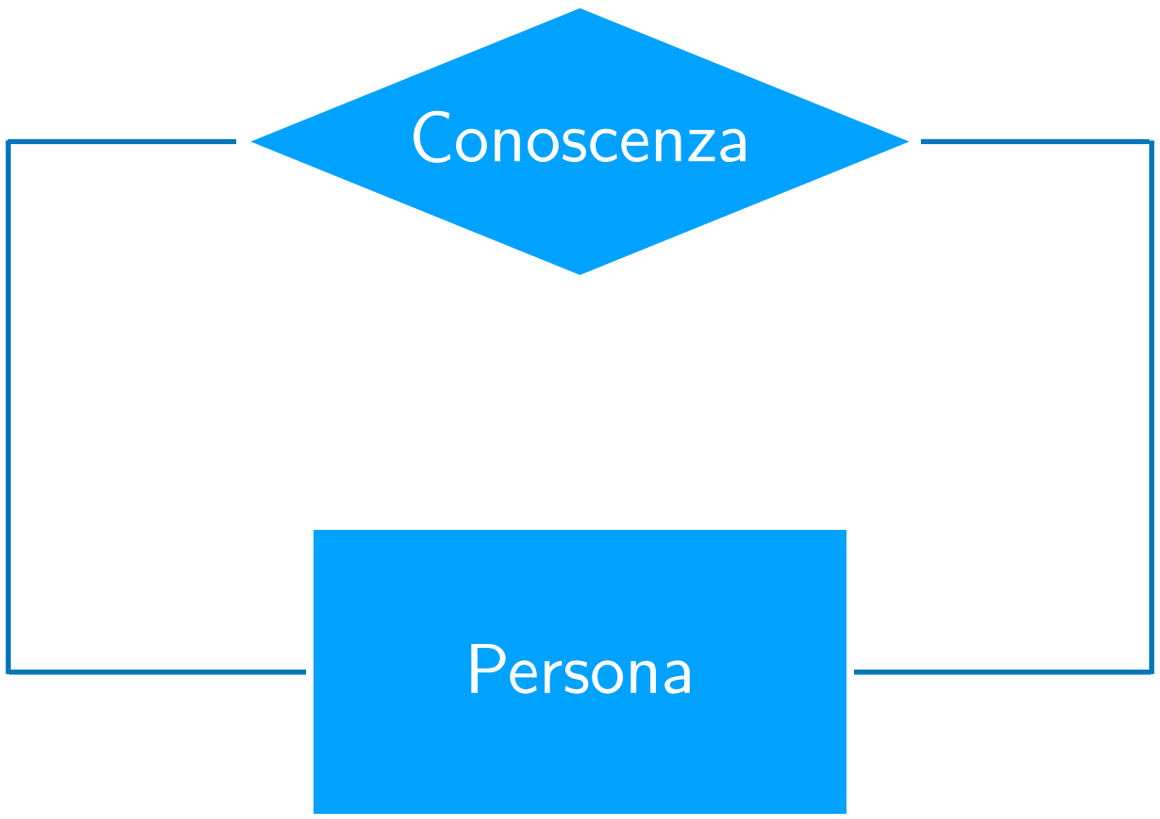
Seconda Rappresentazione



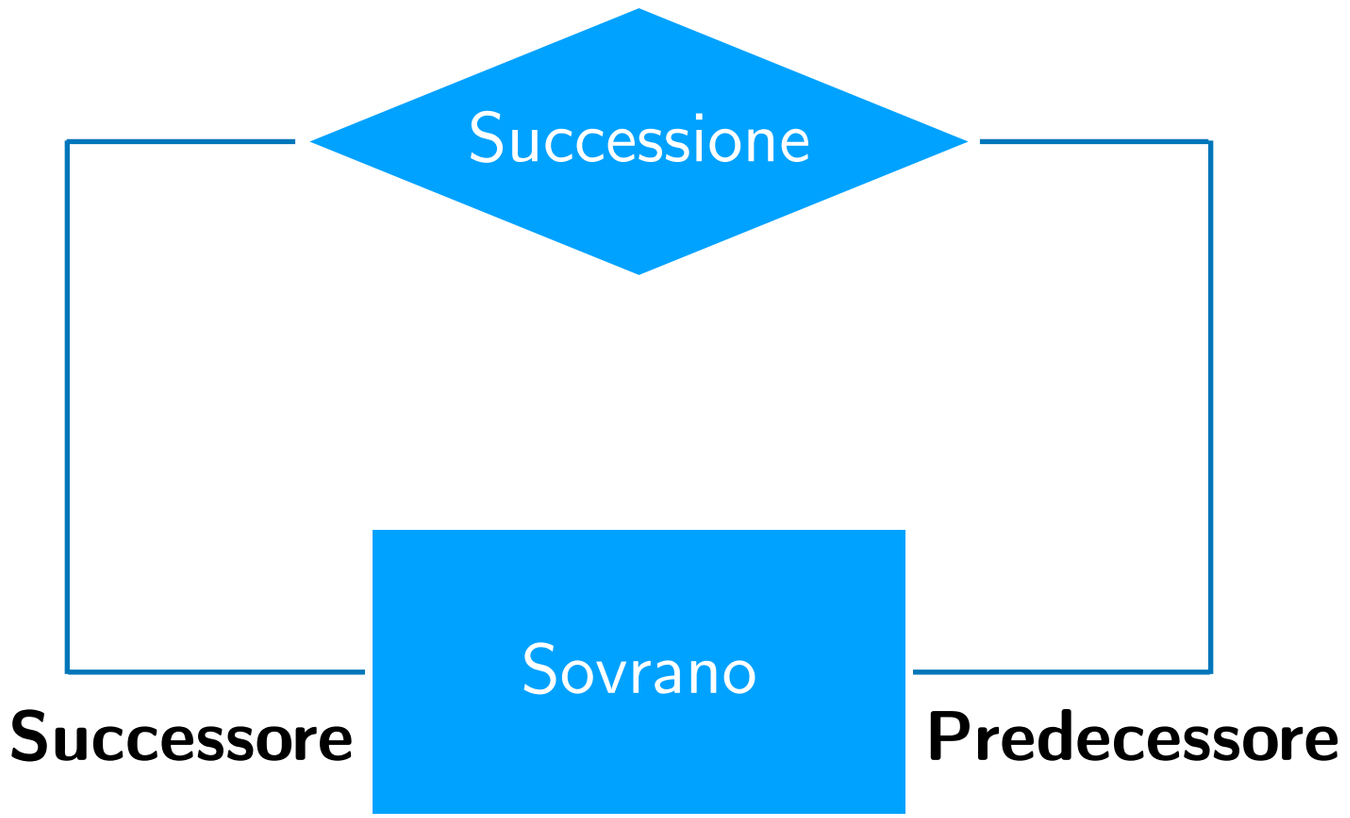
Relationship diverse sulle stesse entità



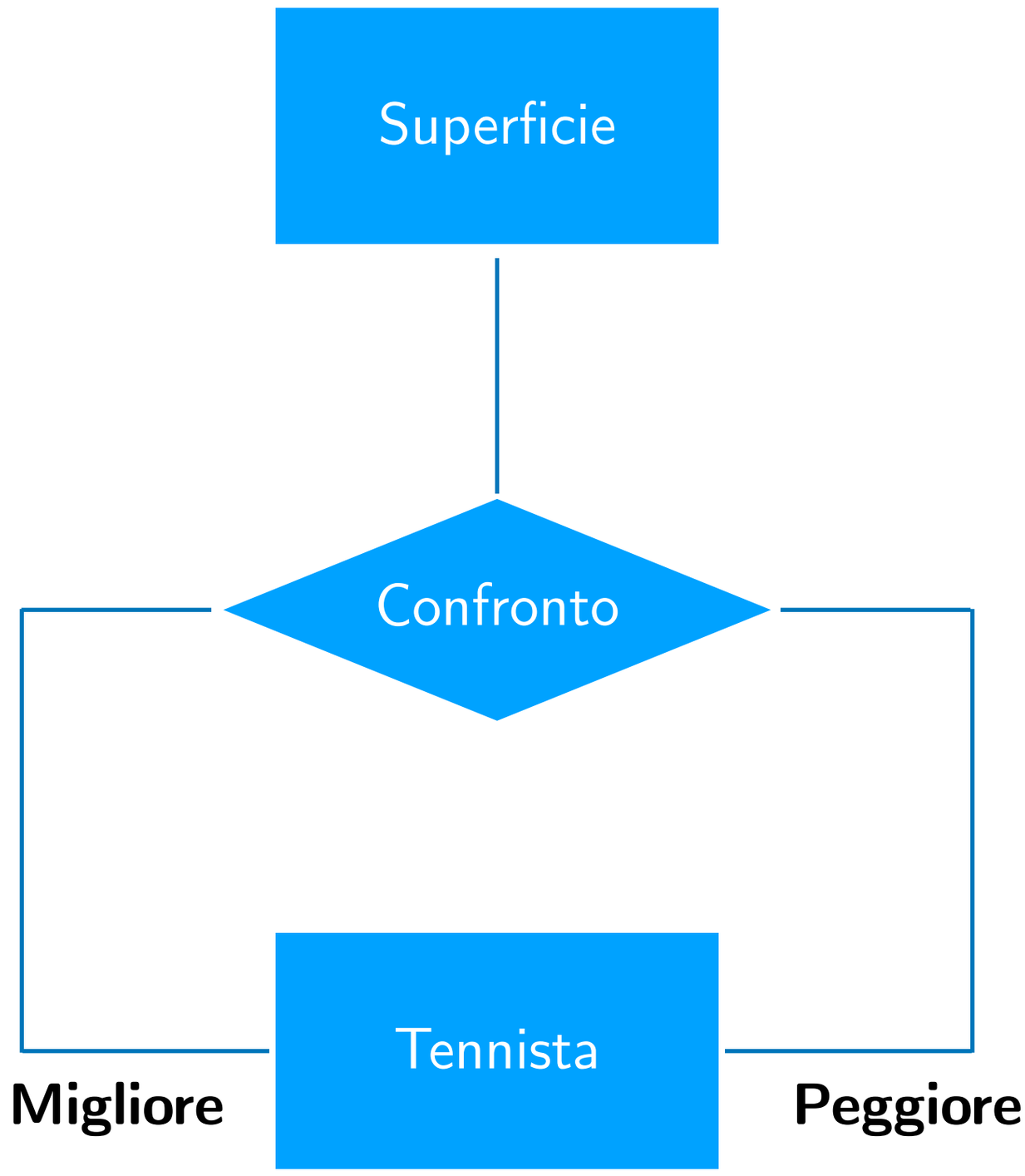
Relationship Ricorsiva



Relationship Ricorsiva con ruoli



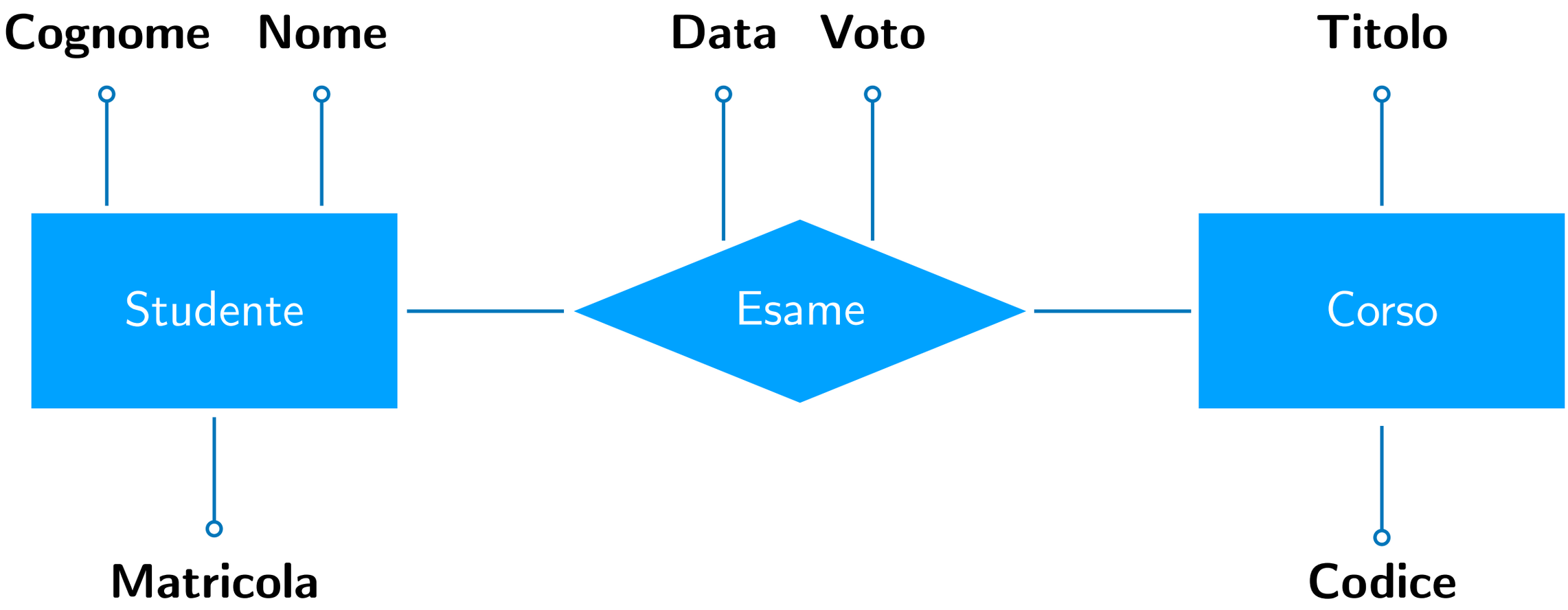
Relationship Mista



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**

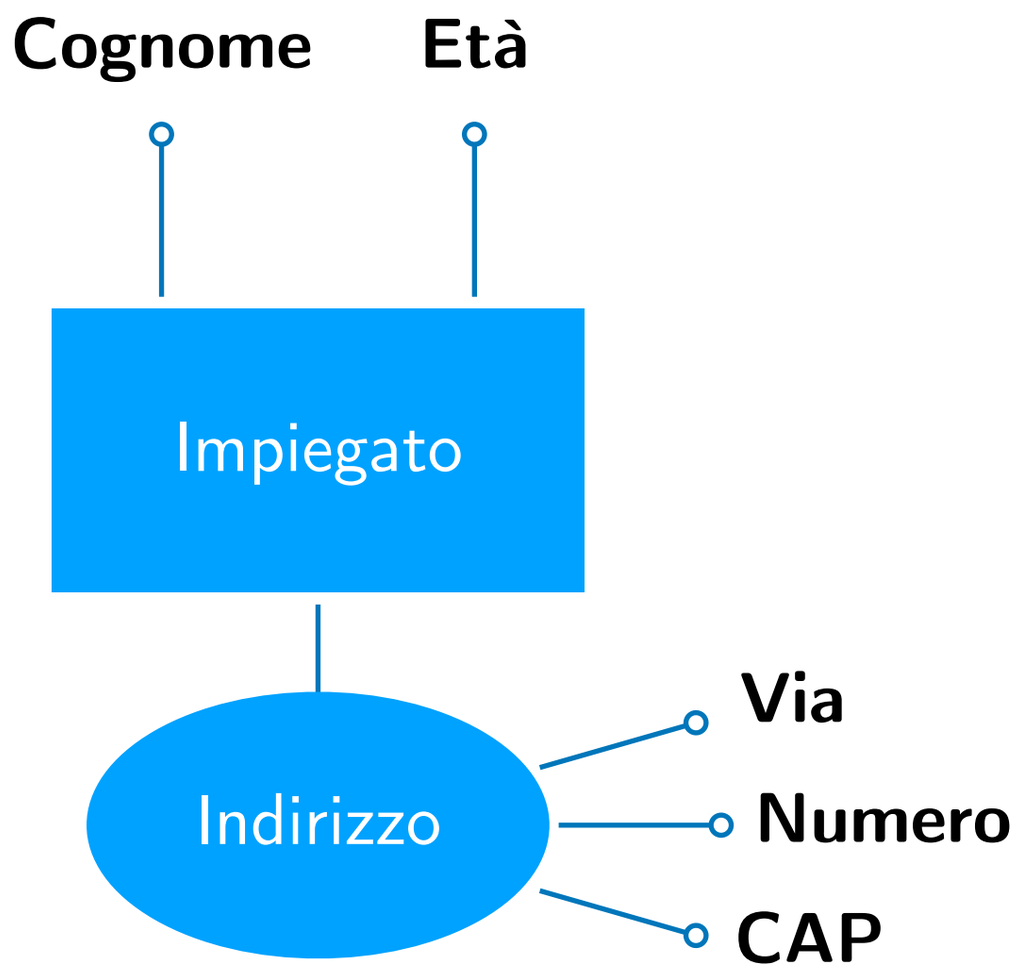
Rappresentazione Grafica di Attributi



Attributi Composti

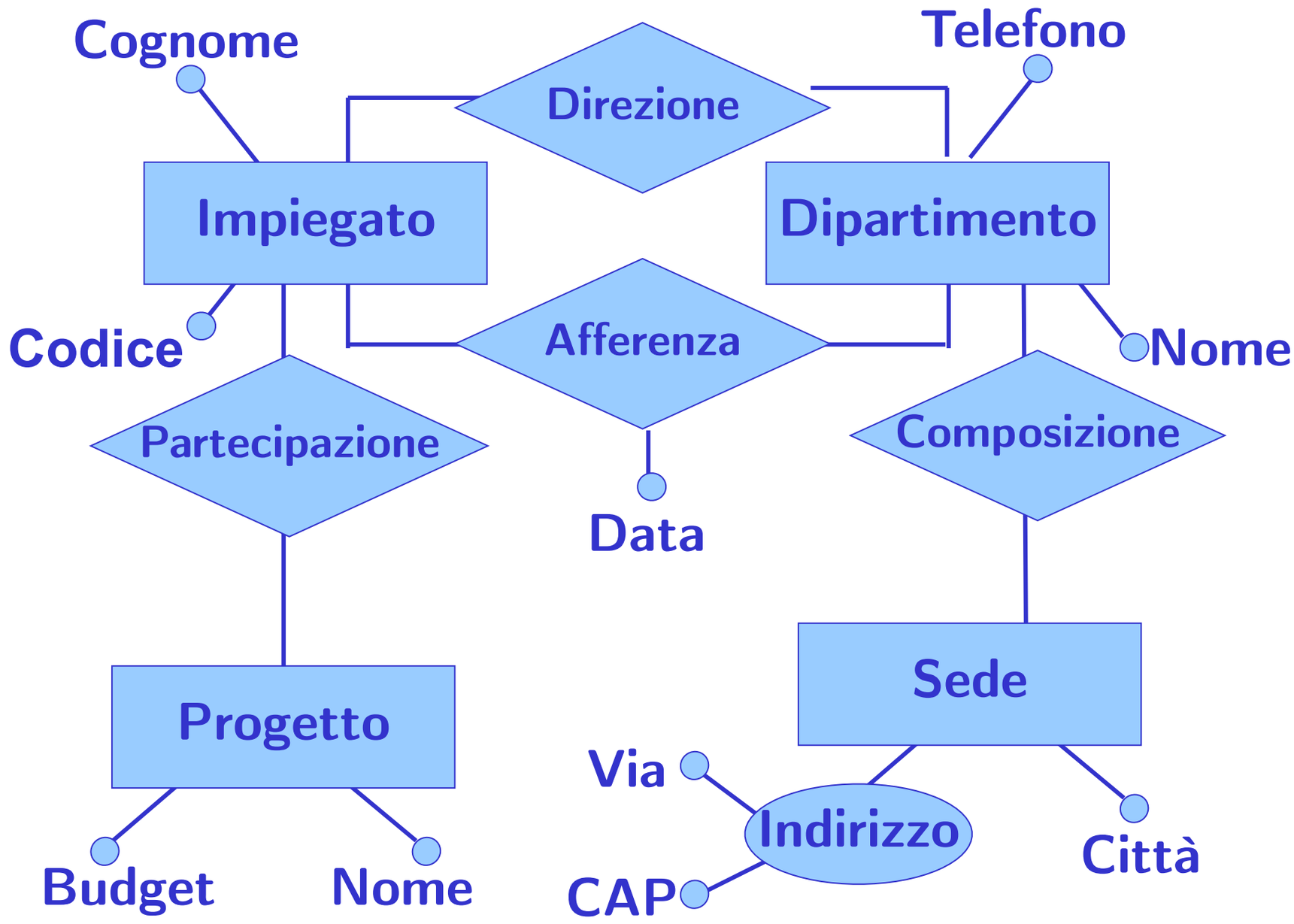
- **Raggruppano** attributi di una **medesima** entità o *relationship* che presentano **affinità** nel loro significato o uso
- Esempio:
  - Via, Numero civico e CAP formano un **Indirizzo**

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


Cognome
Direzione
Telefono
Impiegato
Dipartimento
Codice
Afferenza
Nome
Partecipazione
Composizione
Data
Progetto
Sede
Via
Indirizzo
Città
Budget
Nome
CAP

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**

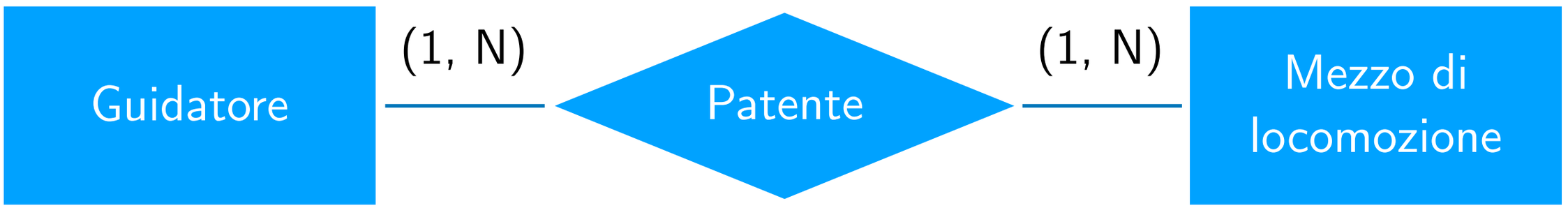
Cardinalità di *relationship*

- 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*

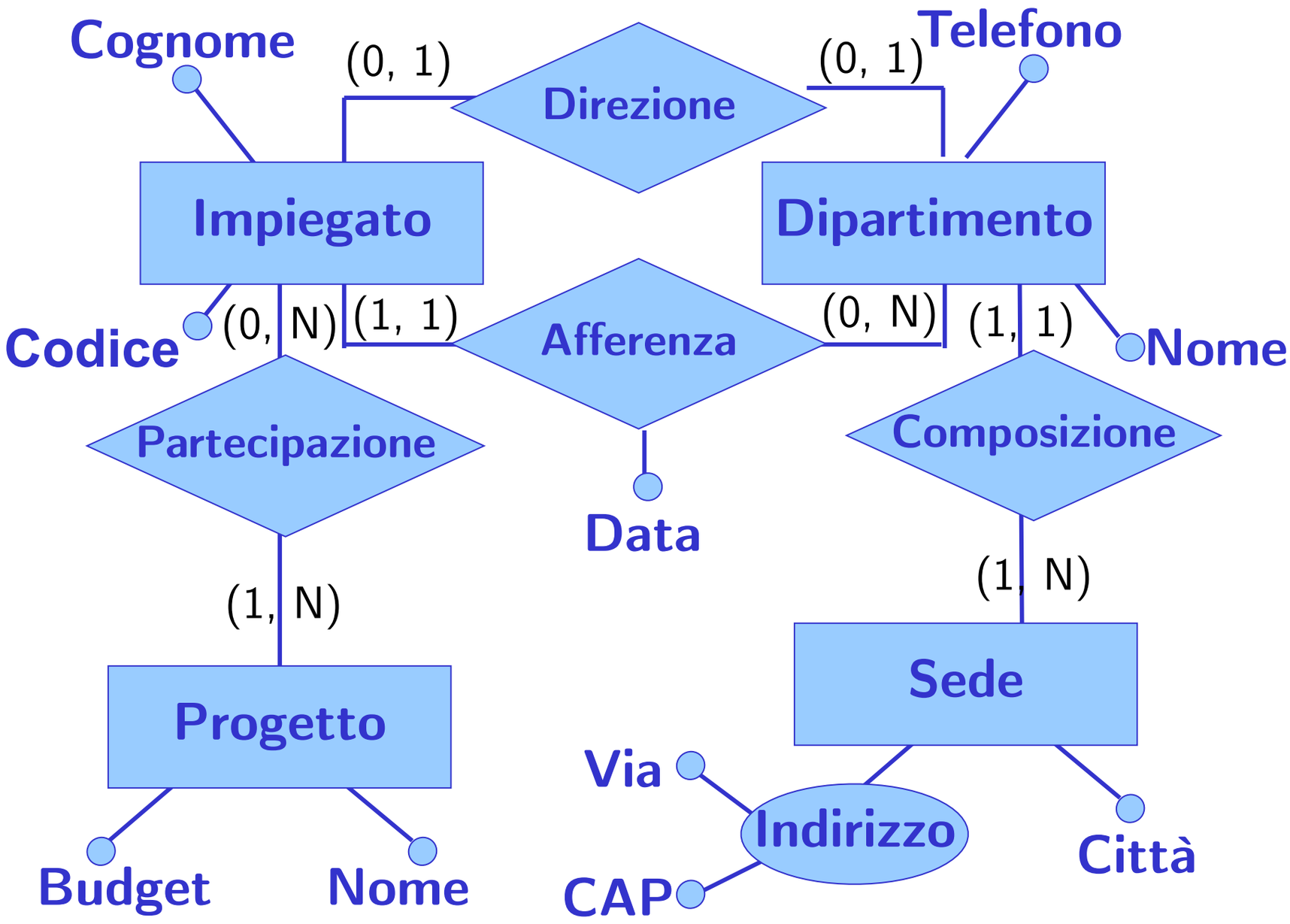
- Con riferimento alle **cardinalità massime**, abbiamo *relationship*:
  - **uno a uno**
  - **uno a molti**
  - **molti a molti**

Relationship "molti a molti"



Relationship "uno a molti"

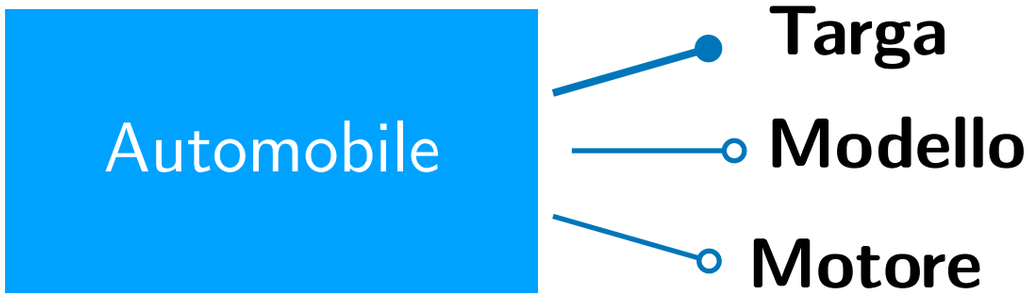
Relationship "uno a uno"

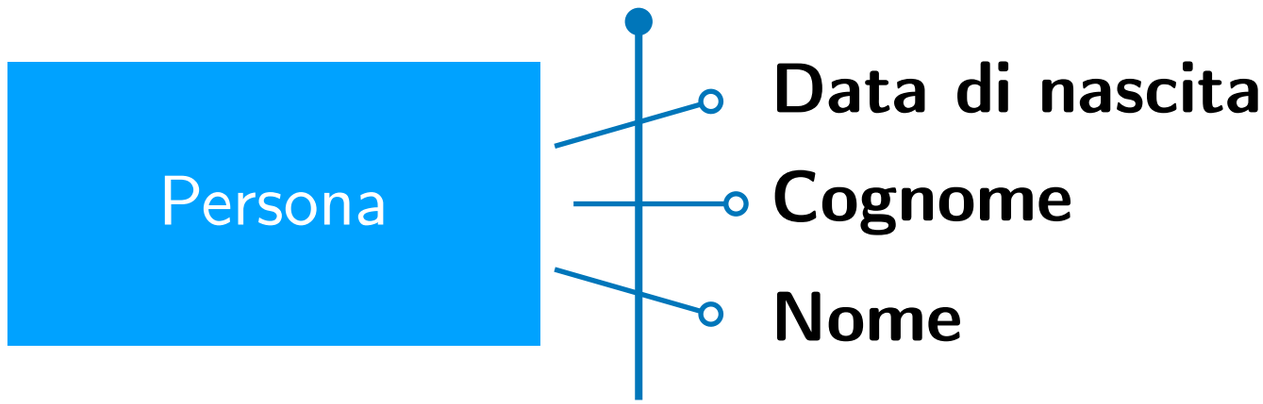

Cognome
(0, 1)
Direzione
(0, 1)
Telefono
Impiegato
Dipartimento
Codice
(0, N)
(1, 1)
Afferenza
(0, N)
(1, 1)
Nome
Partecipazione
Composizione
Data
(1, N)
(1, N)
Progetto
Sede
Via
Indirizzo
Città
CAP
Budget
Nome

Identificatore di entità

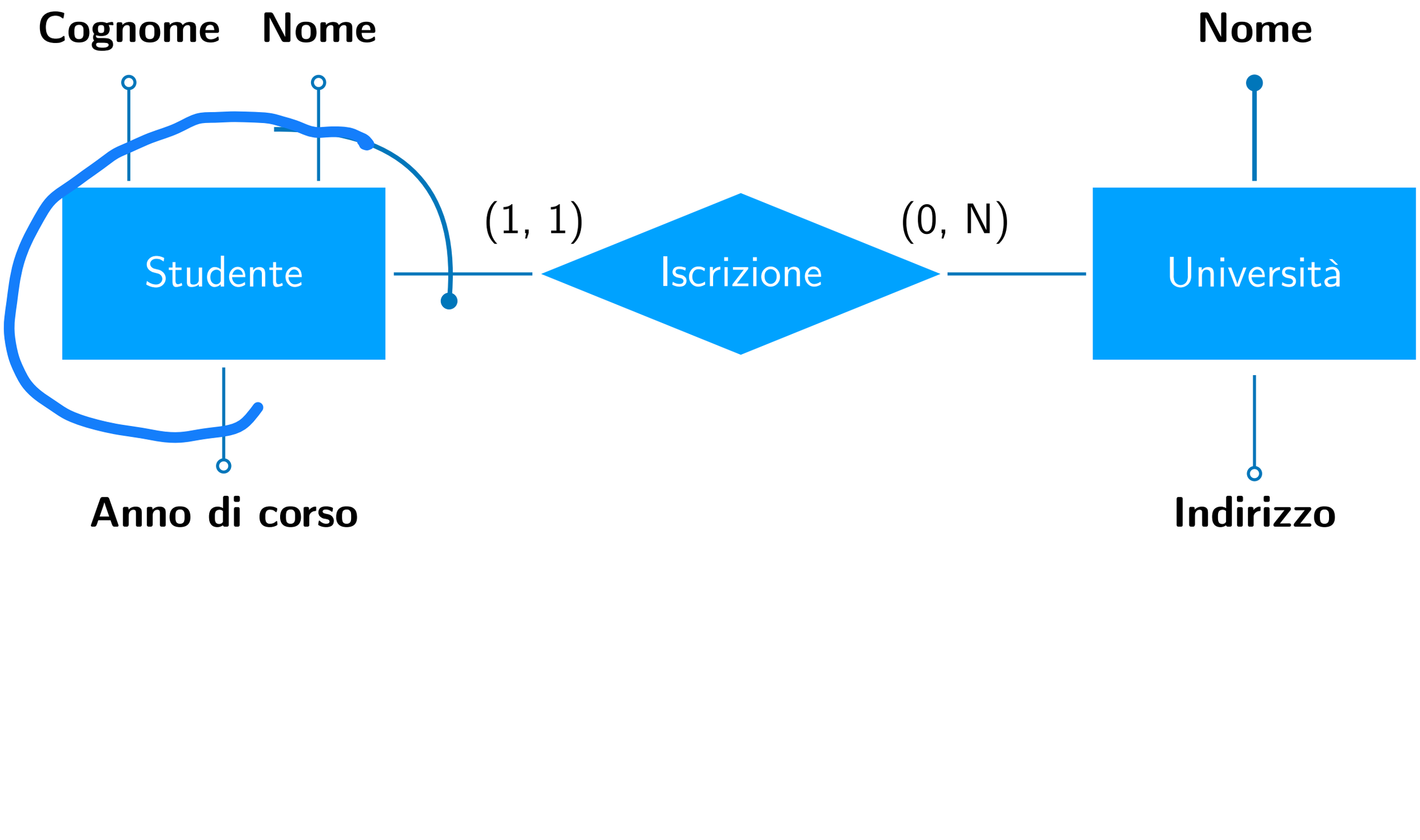
- "Strumento" per l'identificazione univoca delle occorrenze di un'entità
- Costituito da:
  - attributi dell'entità
    - **identificatore interno** (o chiave)
  - (attributi +) l'identificatore interno di entità esterne raggiunta attraverso *relationship*
    - **identificatore esterno**

Identificatori Interni



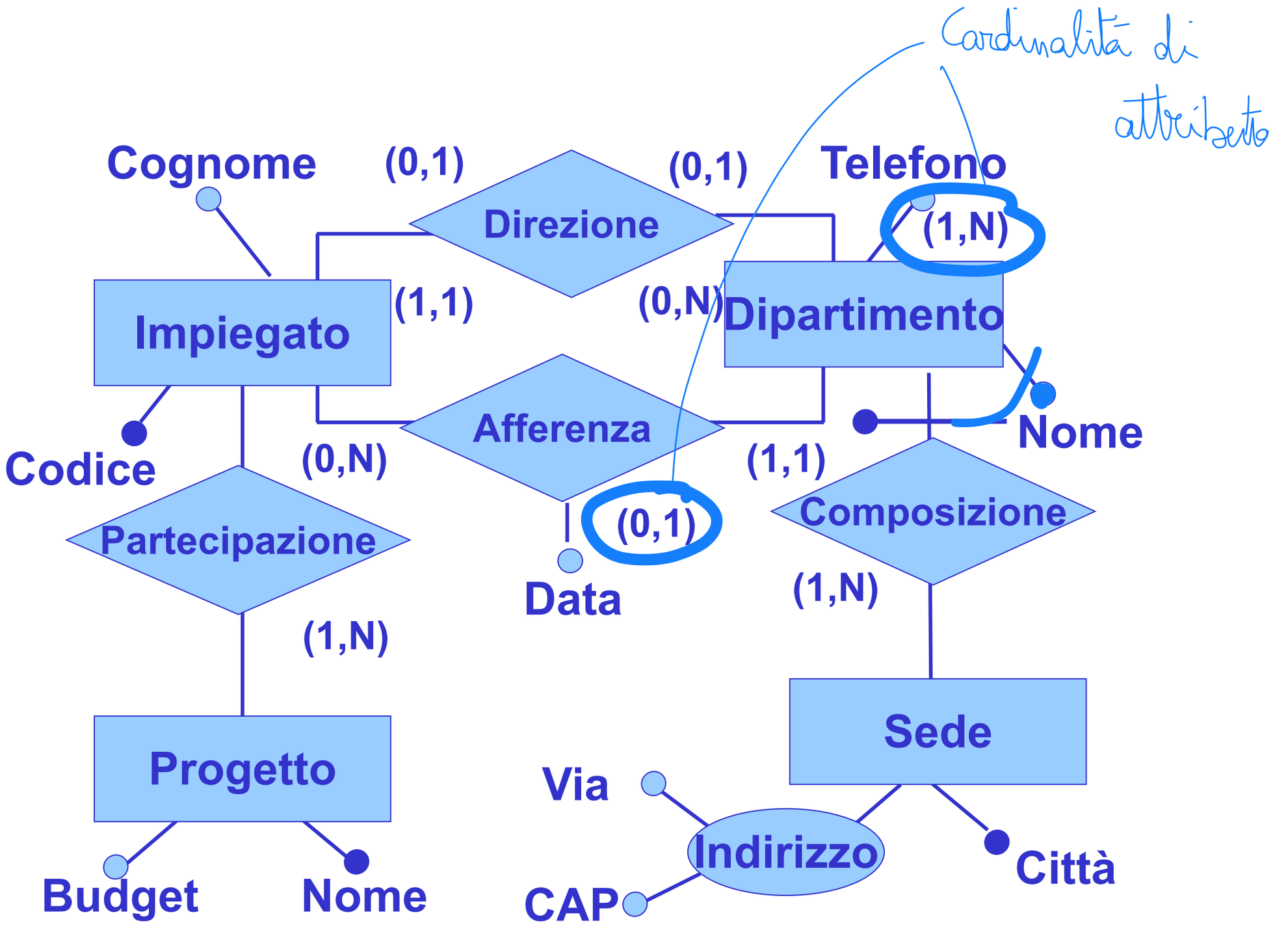


Identificatori Esterni



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)**

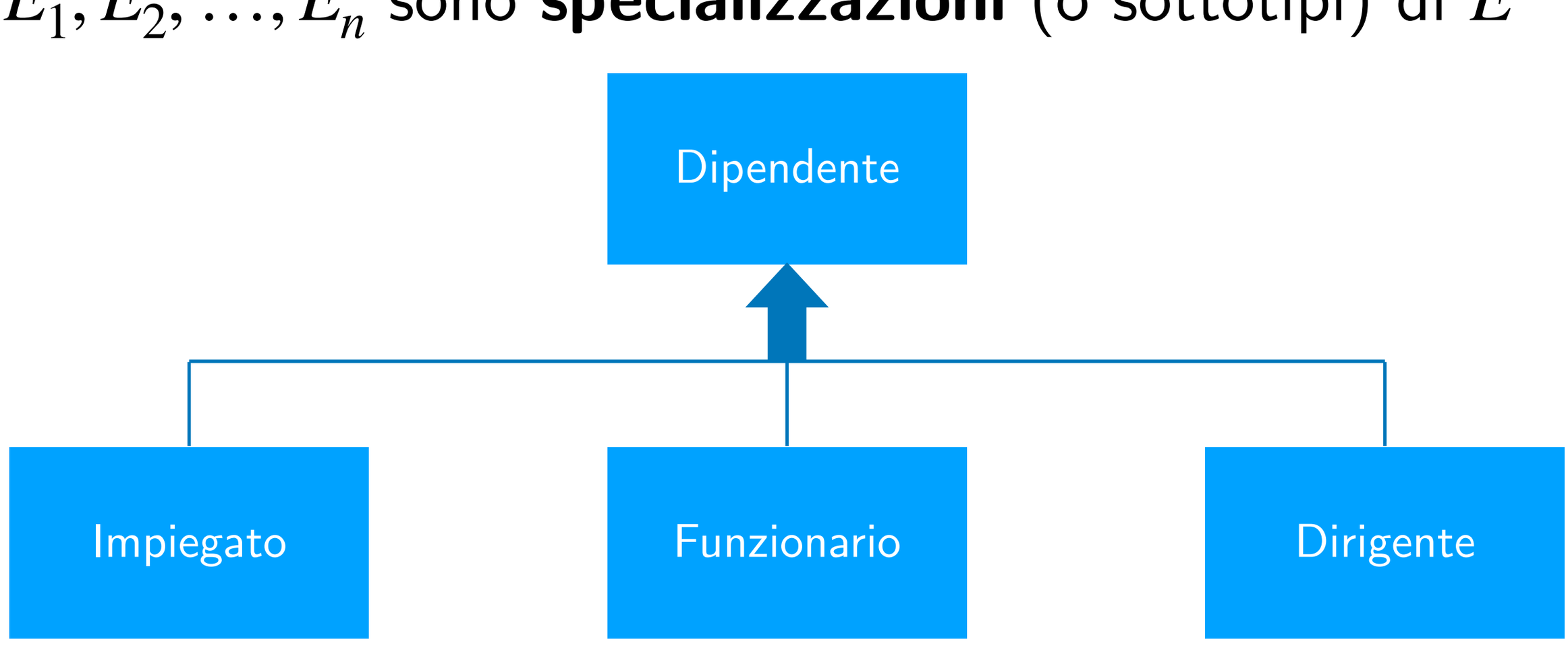

Cardinalità di attributo
Cognome
(0,1)
Direzione
(0,1)
Telefono
(1,N)
Impiegato
(1,1)
(0,N)
Dipartimento
Afferenza
Nome
Codice
(0,N)
(1,1)
Partecipazione
(0,1)
Composizione
Data
(1,N)
(1,N)
Sede
Progetto
Via
Indirizzo
Città
CAP
Budget
Nome

Generalizzazione

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

Generalizzazione

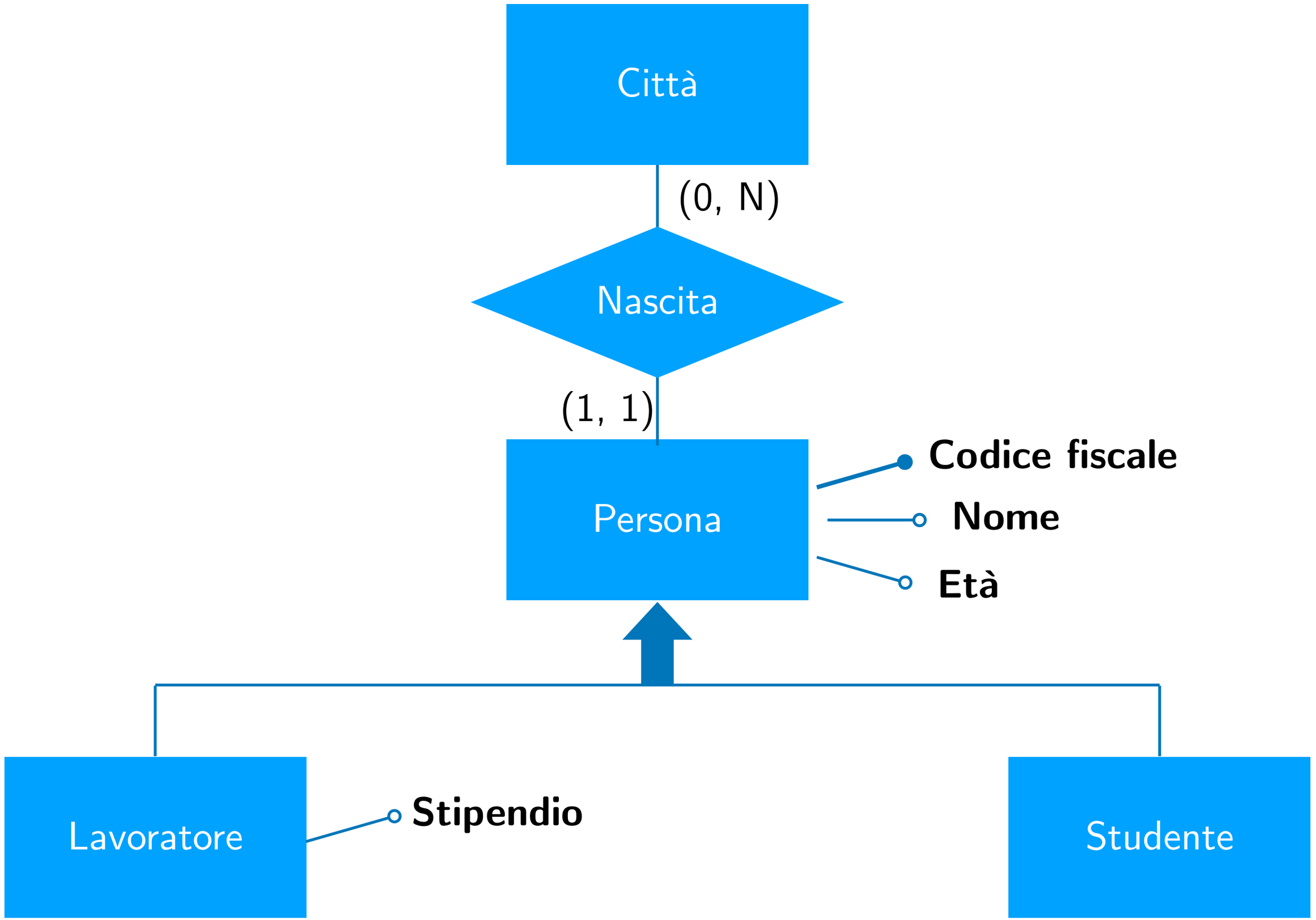
- Mette in relazione **una o più entità** $E_1, E_2, \ldots, E_n$ con una entità E, che le comprende come **casi particolari**
- E è una **generalizzazione** di $E_1, E_2, \ldots, E_n$
- $E_1, E_2, \ldots, E_n$ sono **specializzazioni** (o sottotipi) di E



Proprietà delle generalizzazioni

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

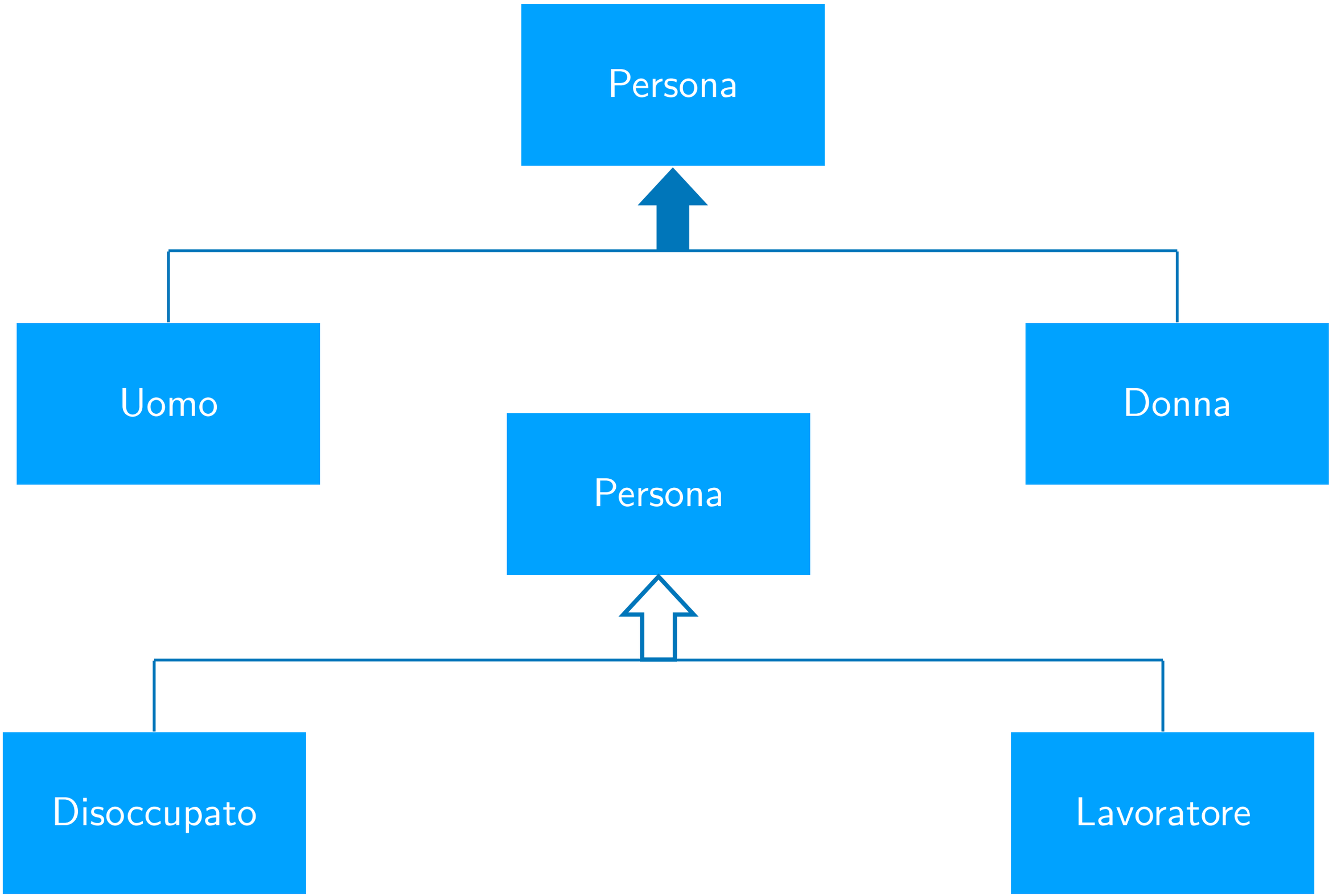
Esempio


Città
(0, N)
Nascita
(1, 1)
Persona
Codice fiscale
Nome
Età
Lavoratore
Stipendio
Studente

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

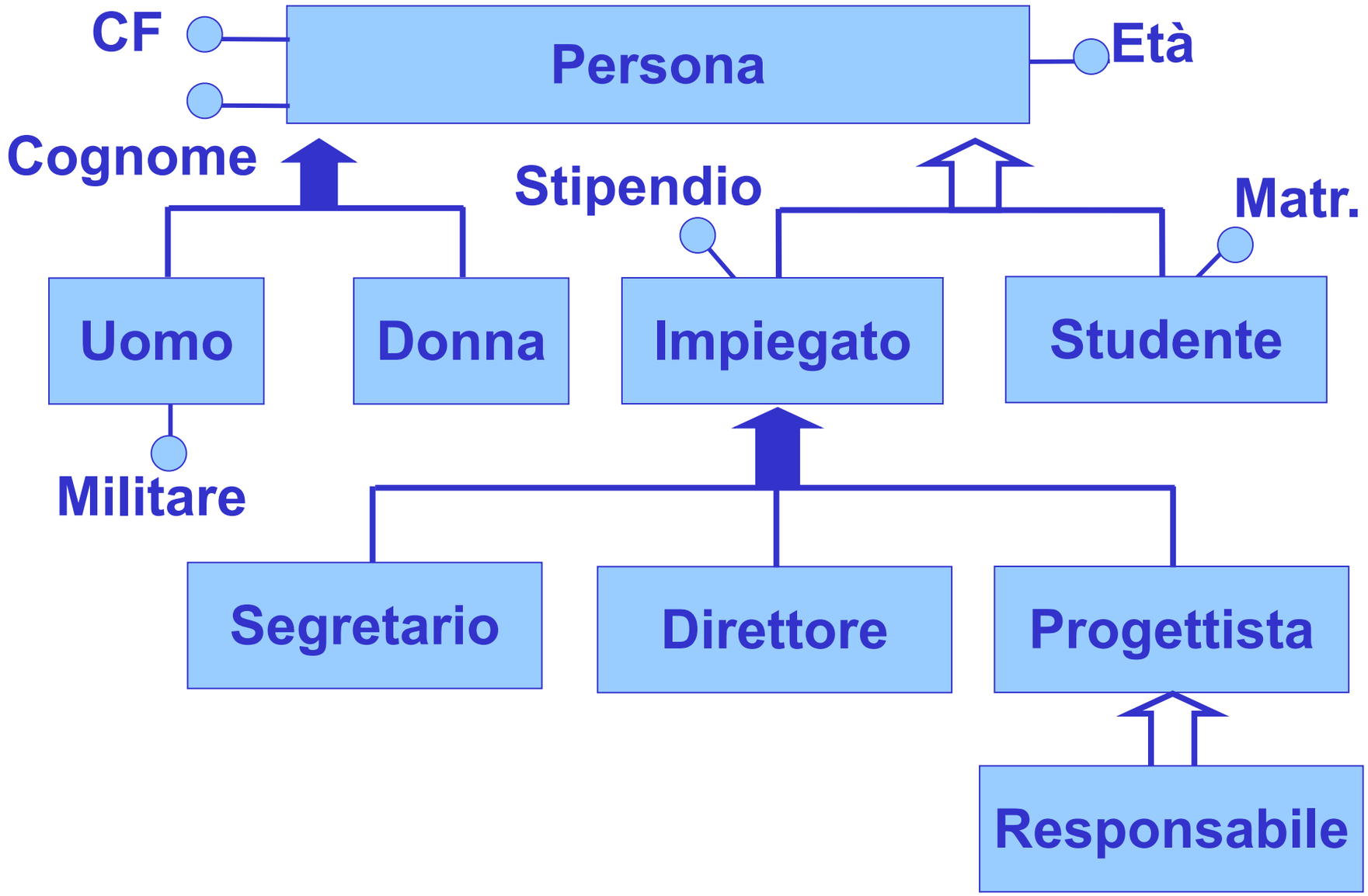


Altre proprietà

- 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 no
- 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)


CF
Persona
Età
Cognome
Stipendio
Matr.
Uomo
Donna
Impiegato
Studente
Militare
Segretario
Direttore
Progettista
Responsabile

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

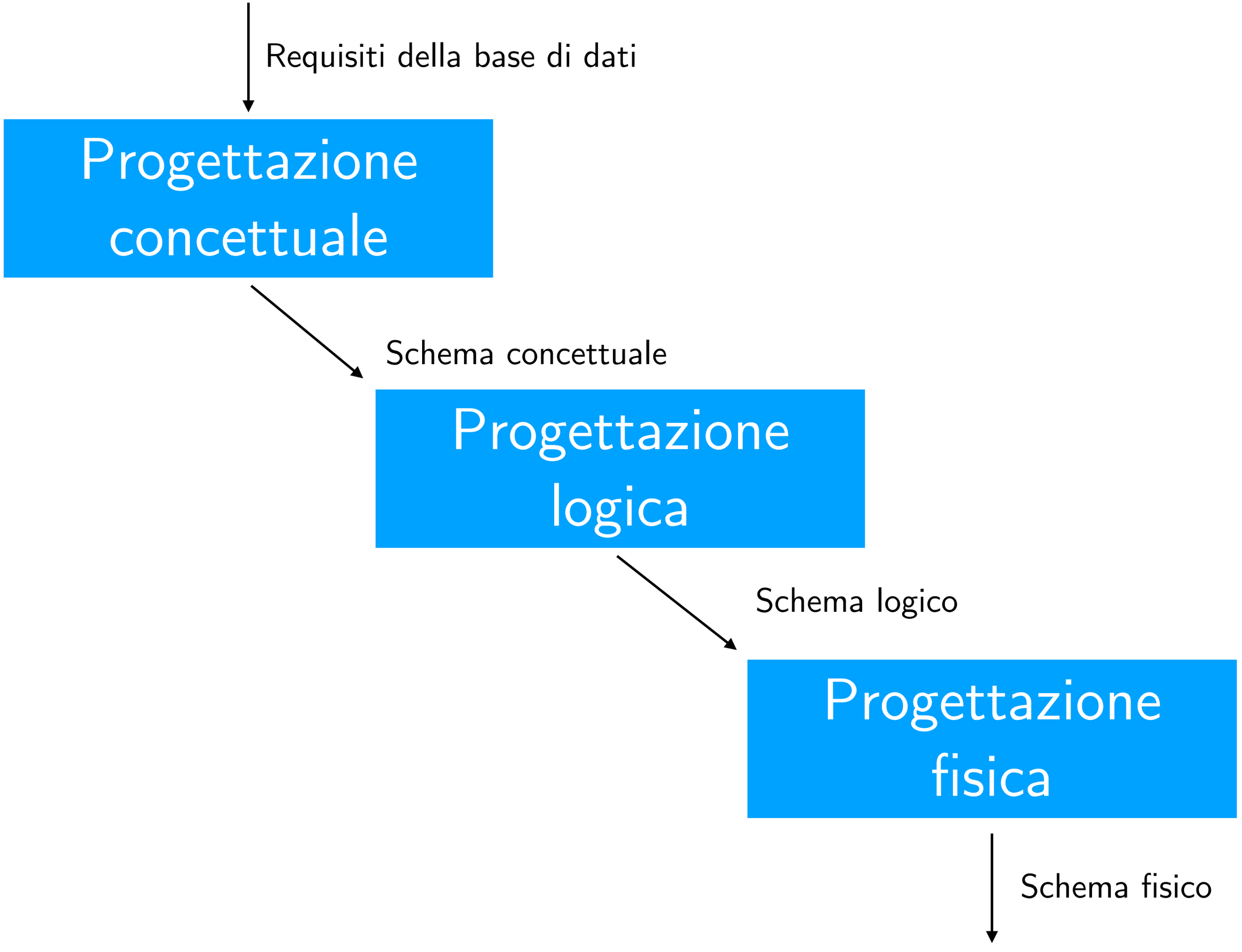
(1) Il direttore di un dipartimento deve a afferire a tale dipartimento
(2) Un impiegato non deve avere uno stipendio maggiore del direttore del dipartimento al quale afferisce
(3) Un dipartimento con sede a Roma deve essere diretto da un impiegato con più di dieci anni di anzianità
(4) Un impiegato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

Progettazione Concettuale

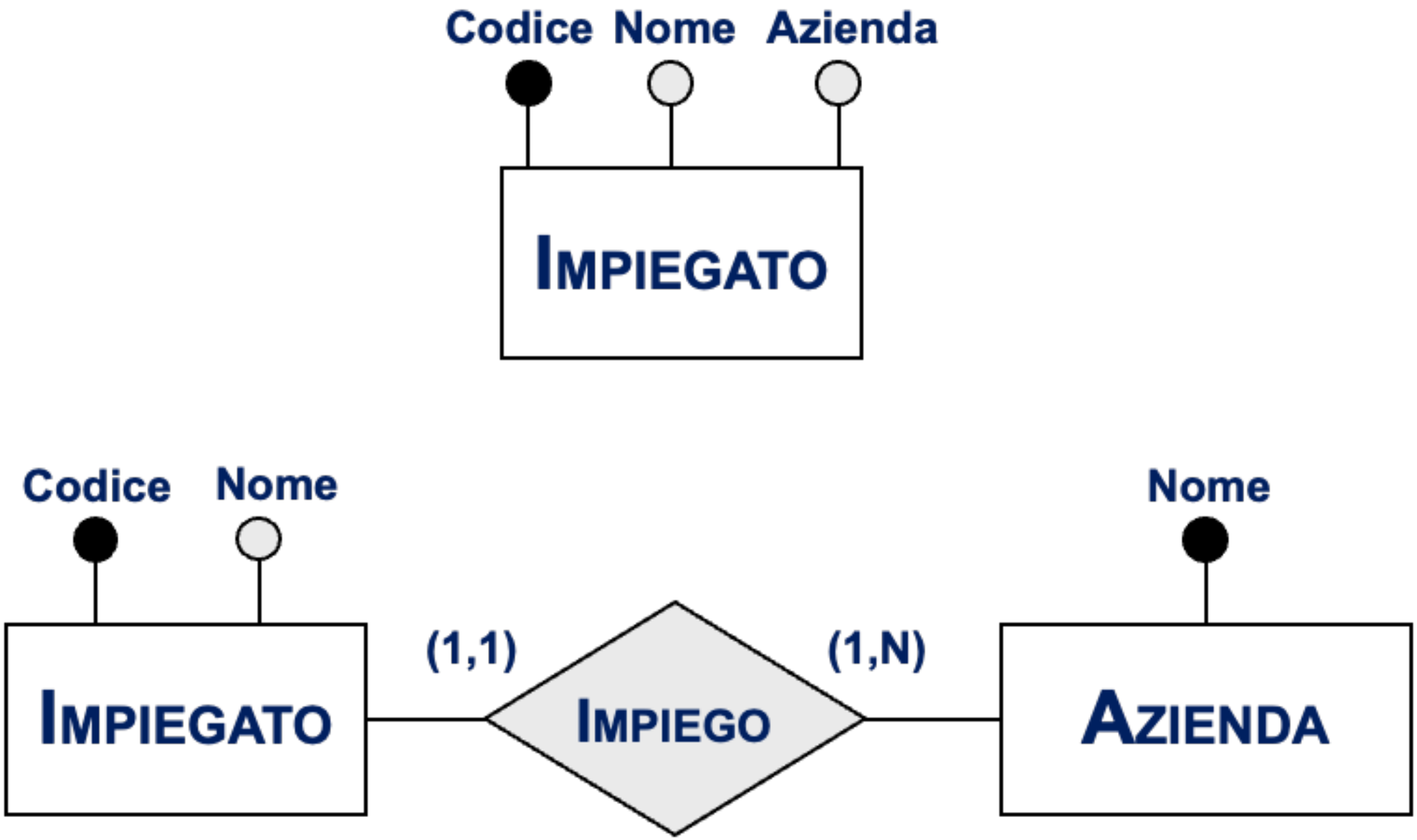

Requisiti della base di dati
Progettazione concettuale
Schema concettuale
Progettazione logica
Schema logico
Progettazione fisica
Schema fisico

- **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**

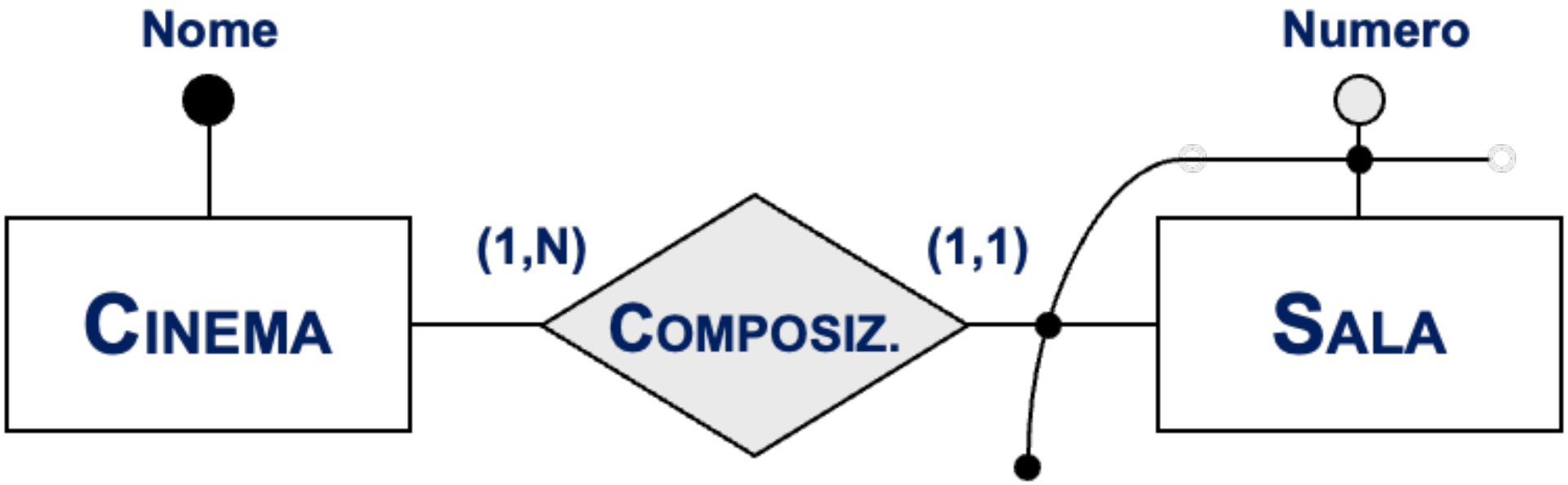
Design Pattern

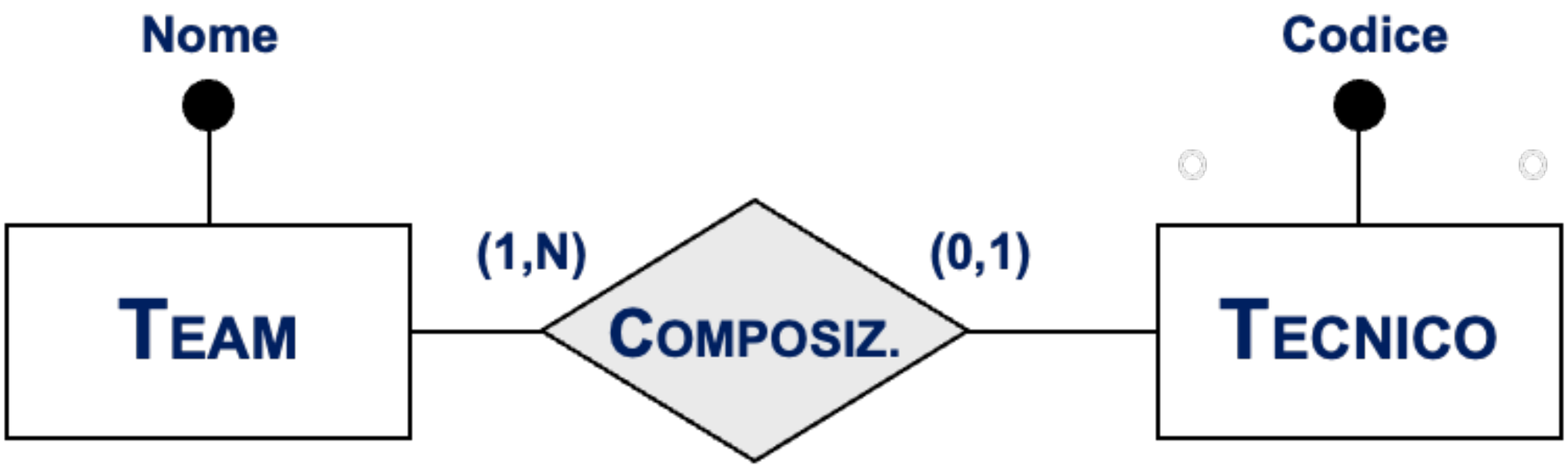
- Soluzioni progettuali a problemi comuni
- Largamente usati nell'ingegneria del software
- Vediamo alcuni pattern comuni nella progettazione concettuale di basi di dati

Reificazione di attributo di entità

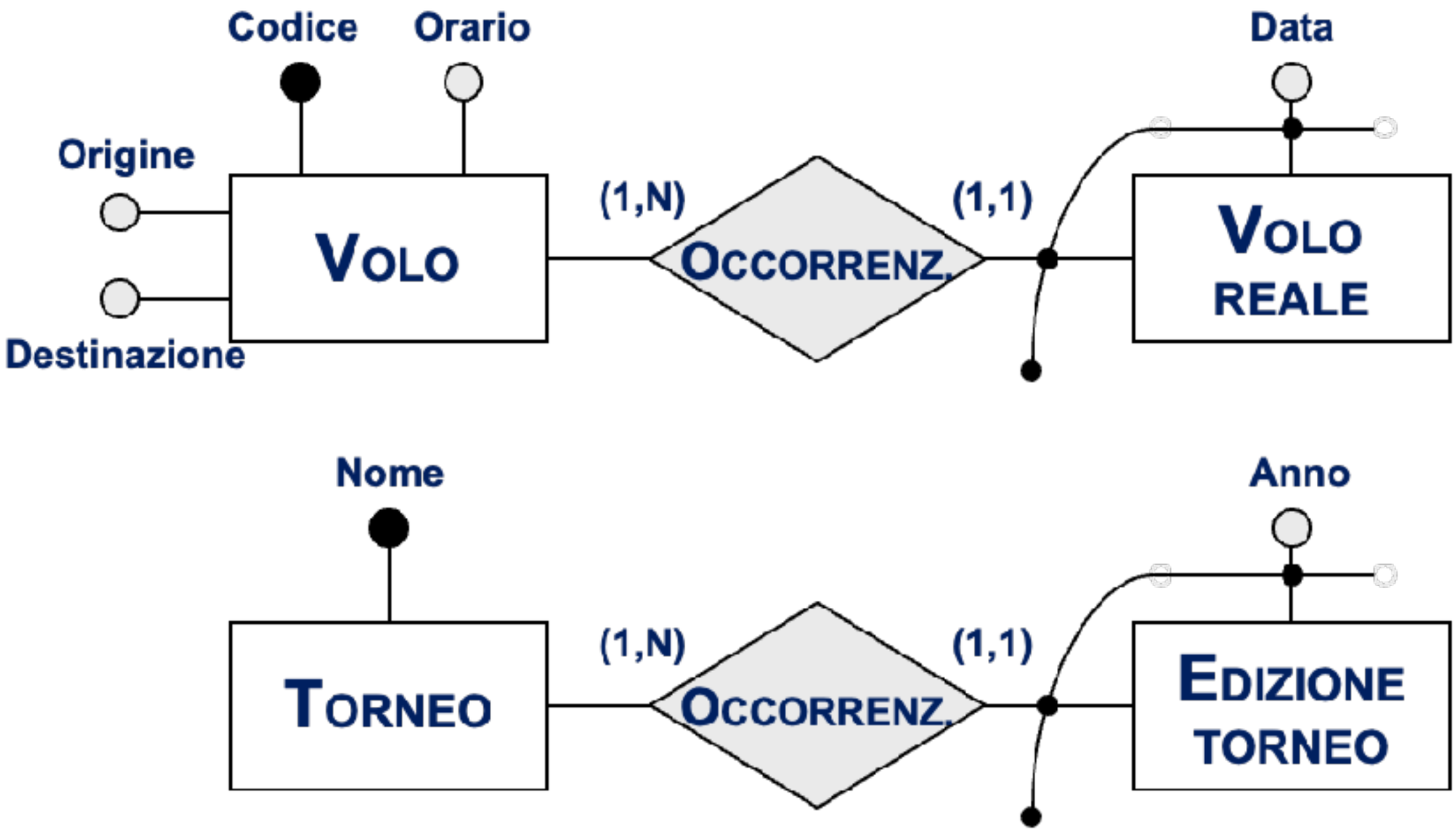


Parte-di

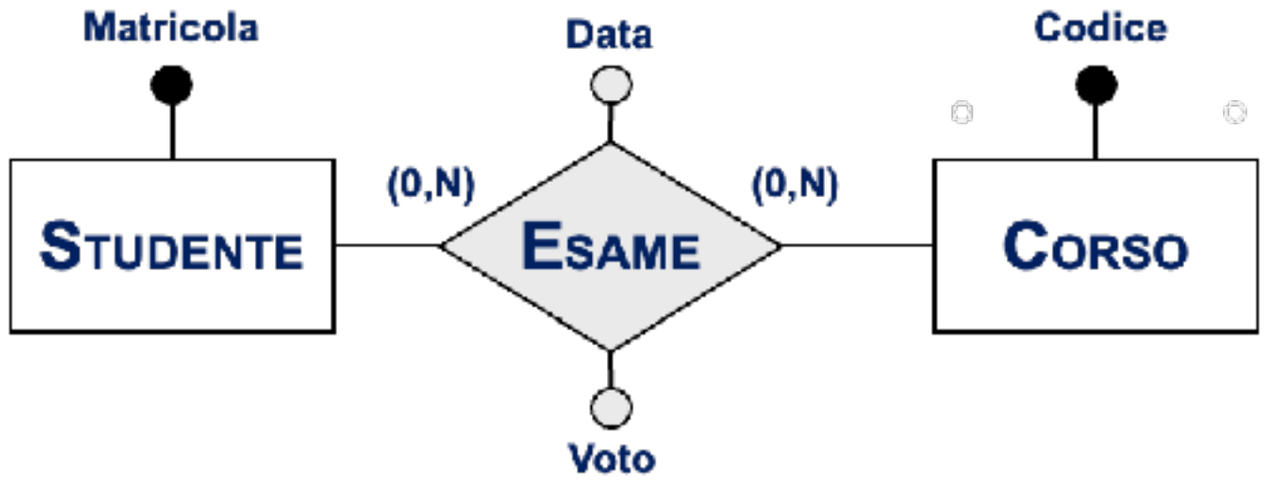

Nome
Numero
CINEMA
(1,N)
COMPOSIZ.
(1,1)
SALA

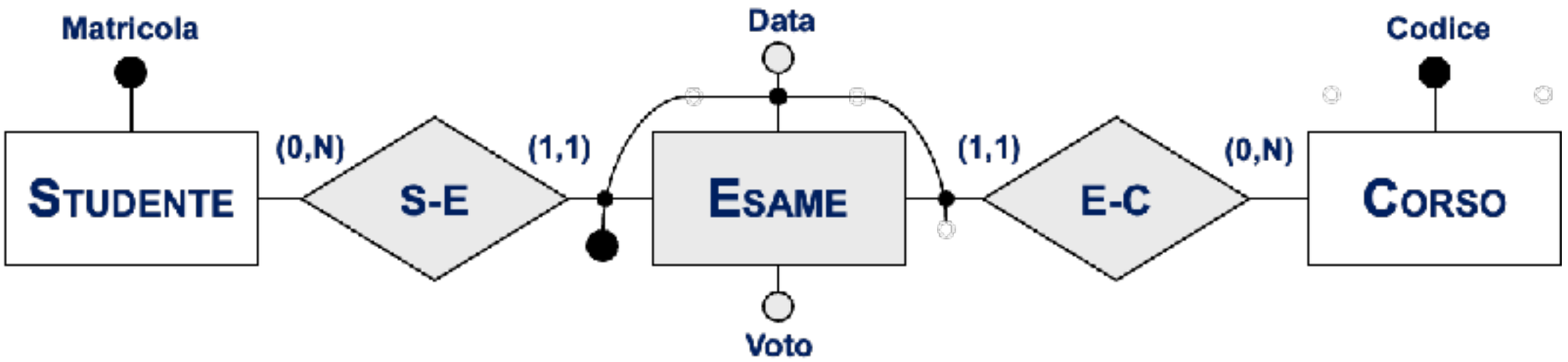

Nome
Codice
TEAM
(1,N)
COMPOSIZ.
(0,1)
TECNICO

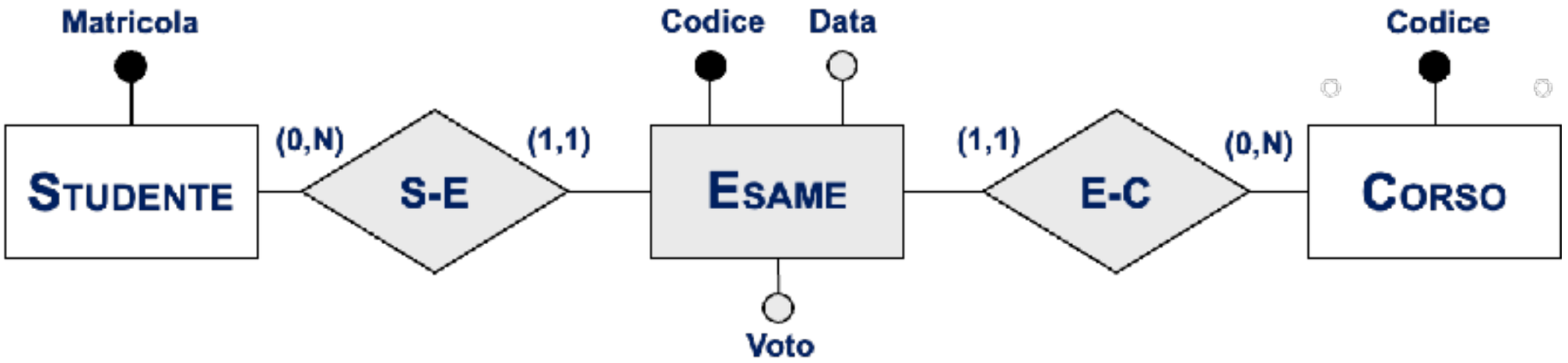
Istanza-di


Codice
Orario
Data
Origine
(1,N)
(1,1)
VOLO
OCCORRENZ.
VOLO REALE
Destinazione
Nome
Anno
TORNEO
(1,N)
(1,1)
OCCORRENZ.
EDIZIONE TORNEO

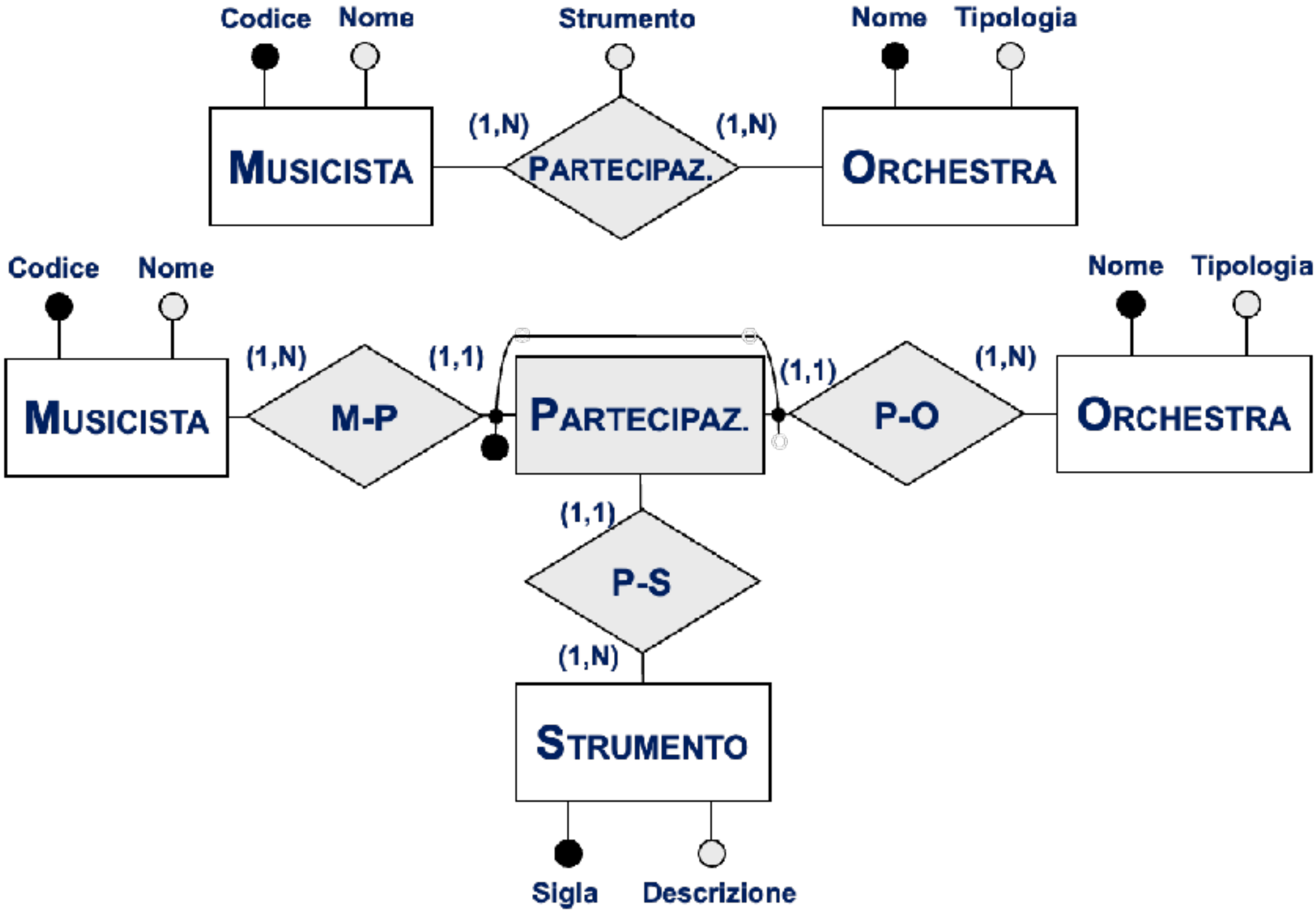
Reificazione di *relationship* binaria



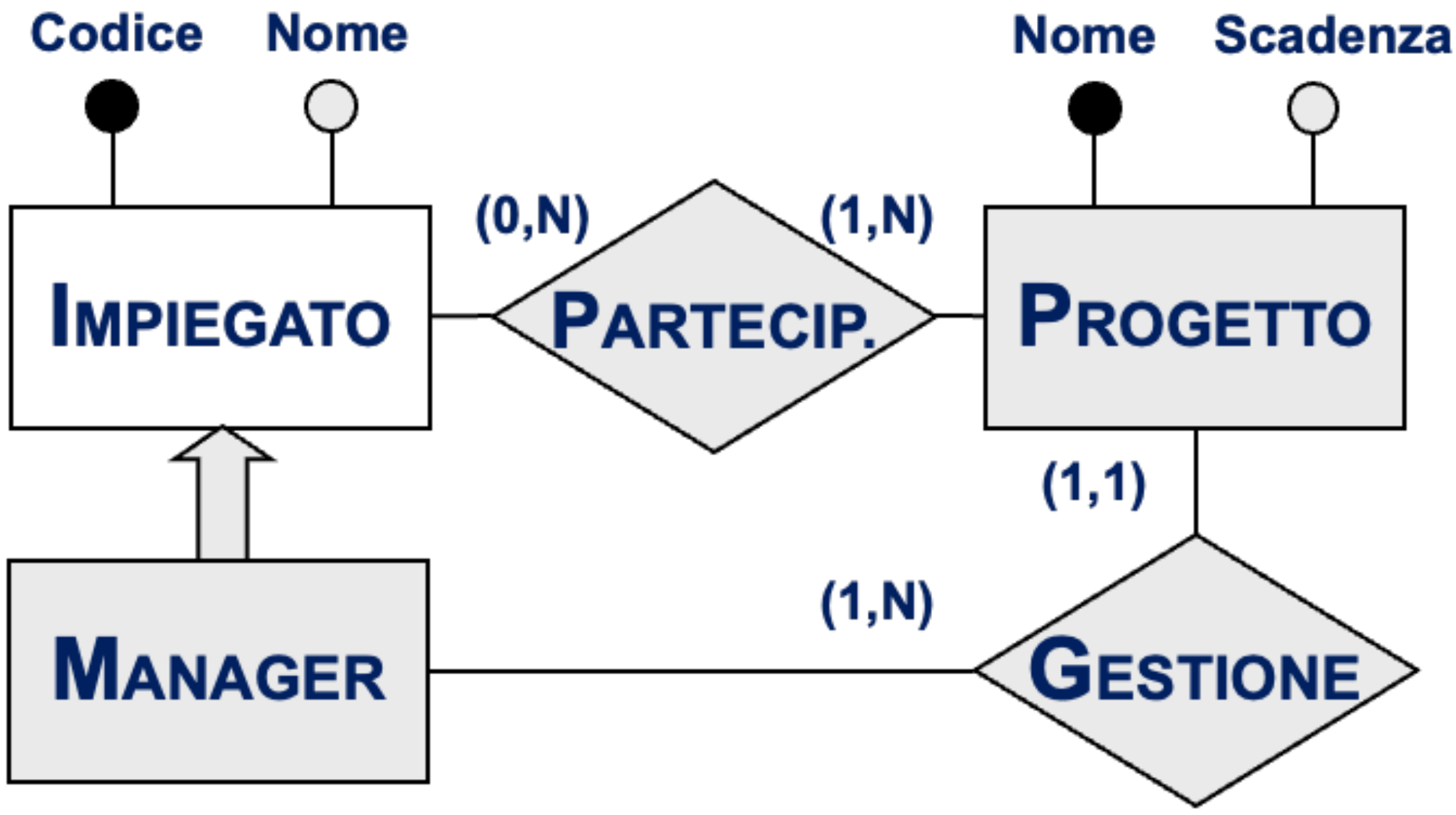




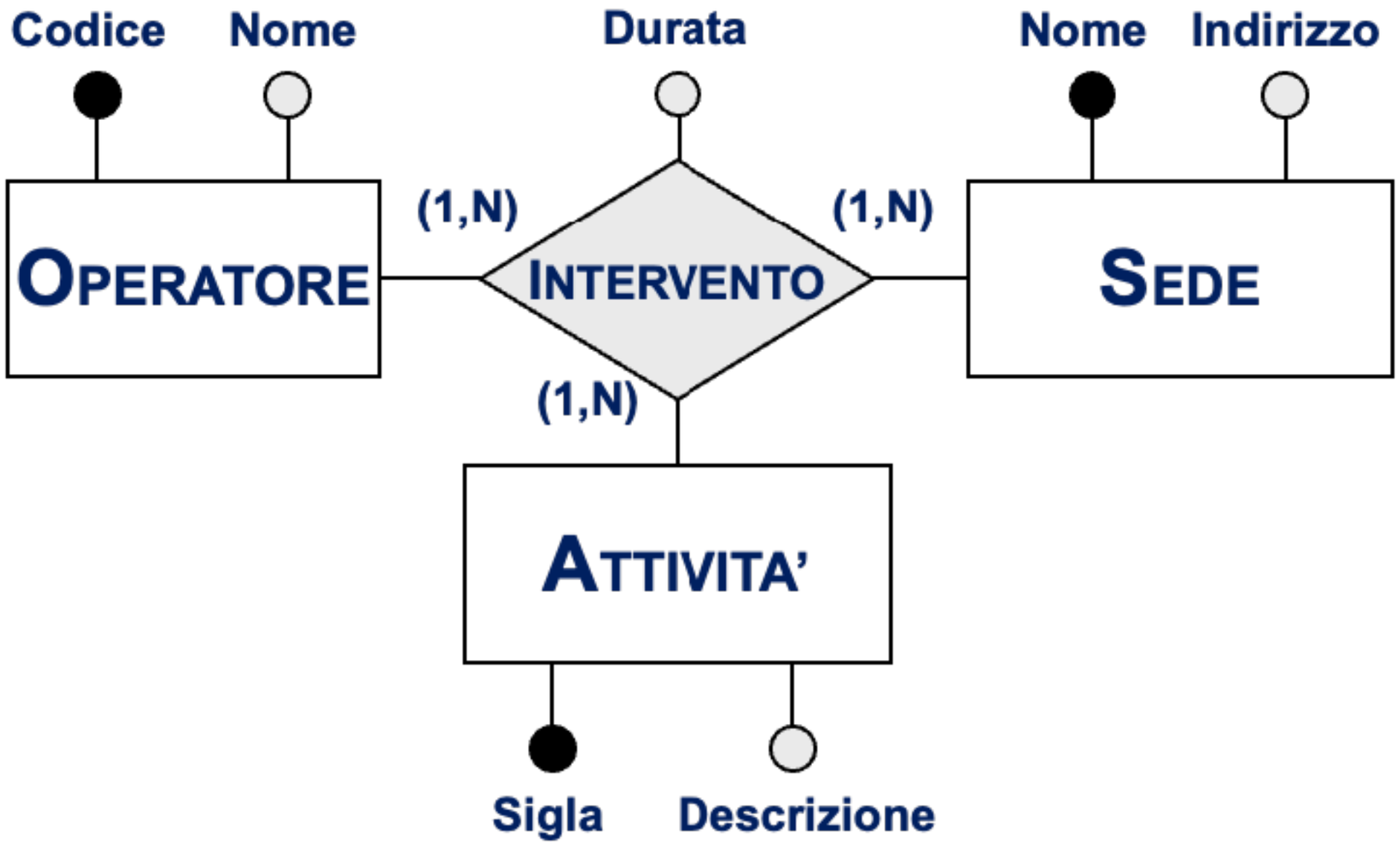
Reificazione di attributo di *relationship*



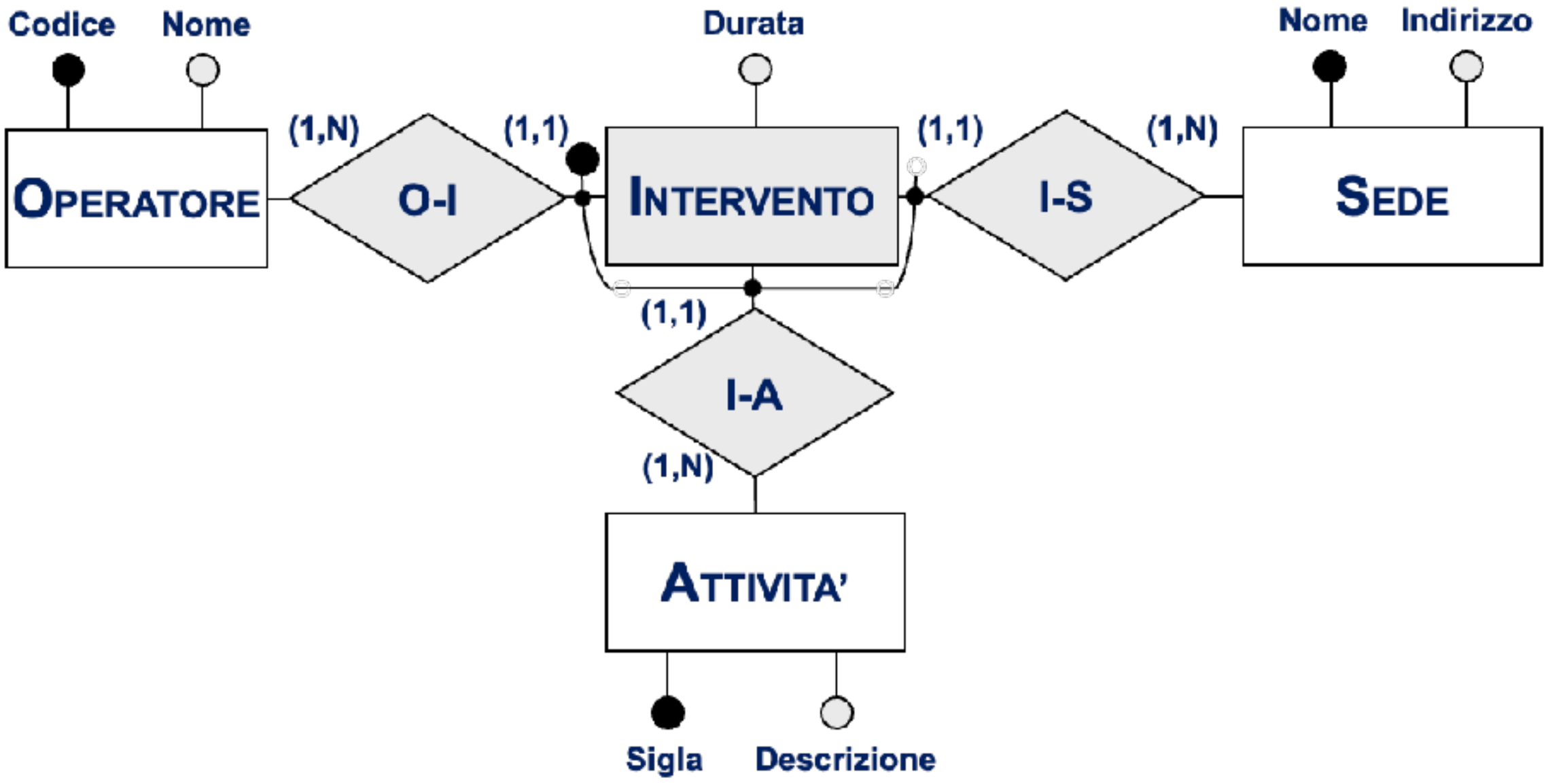
Caso particolare di entità



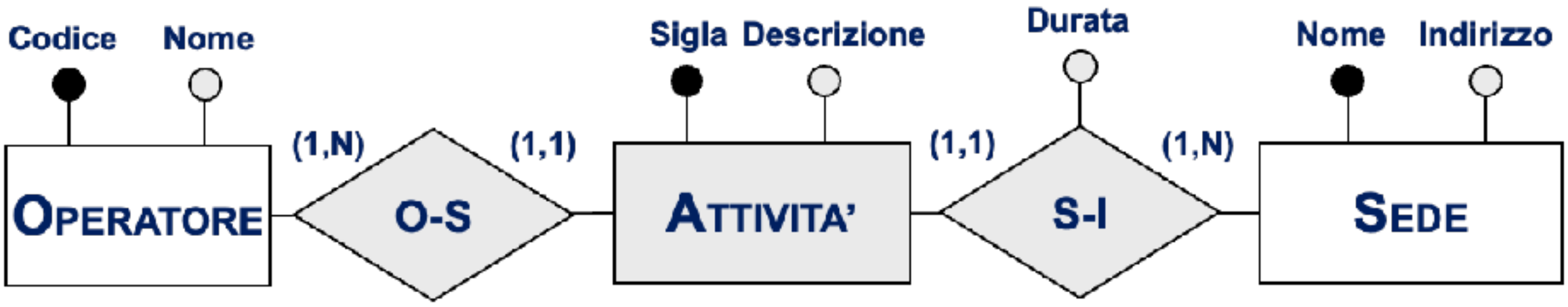
Relationship ternaria



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

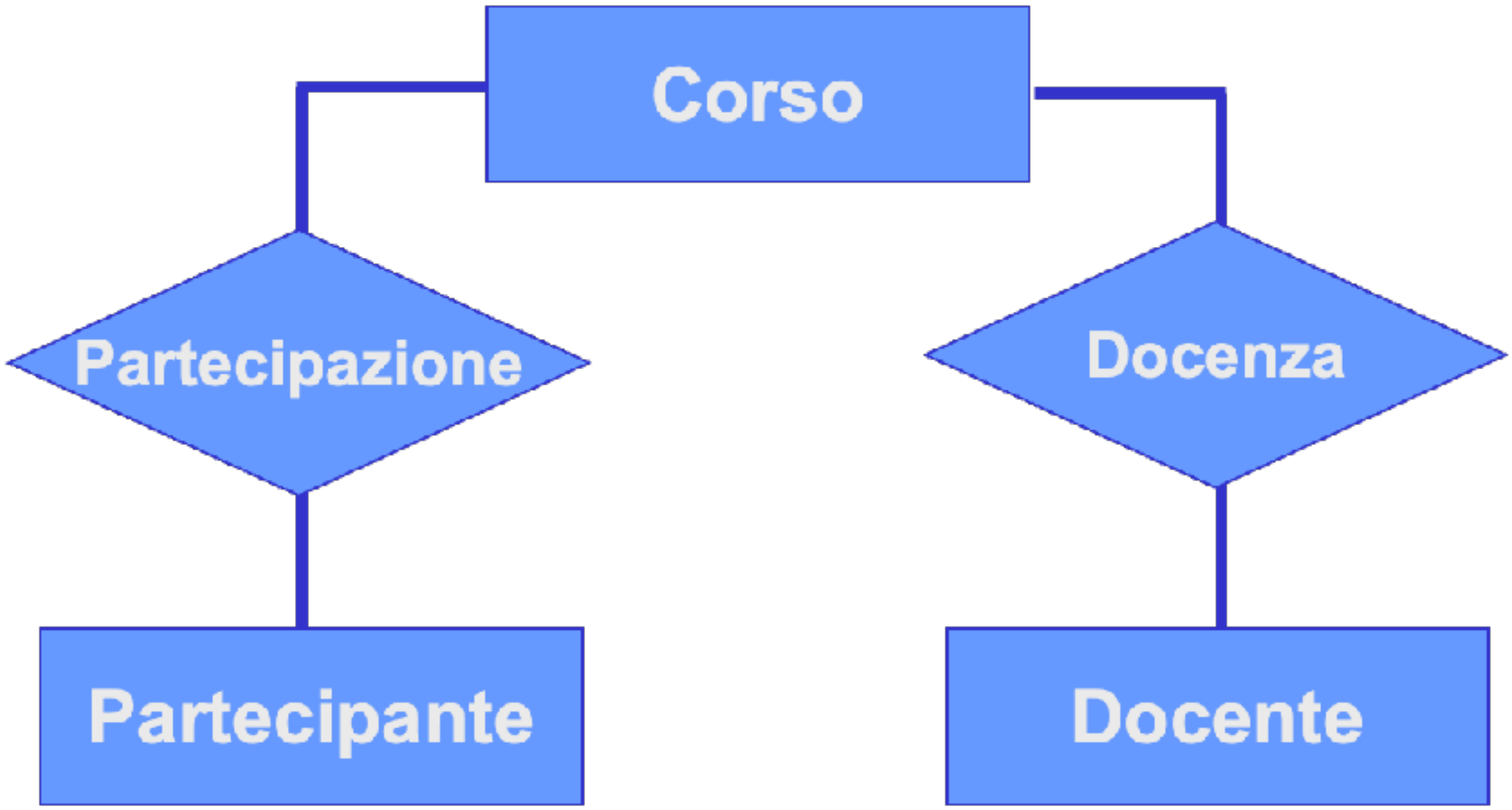
- correttezza
- completezza
- leggibilità
- **minimalità**

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



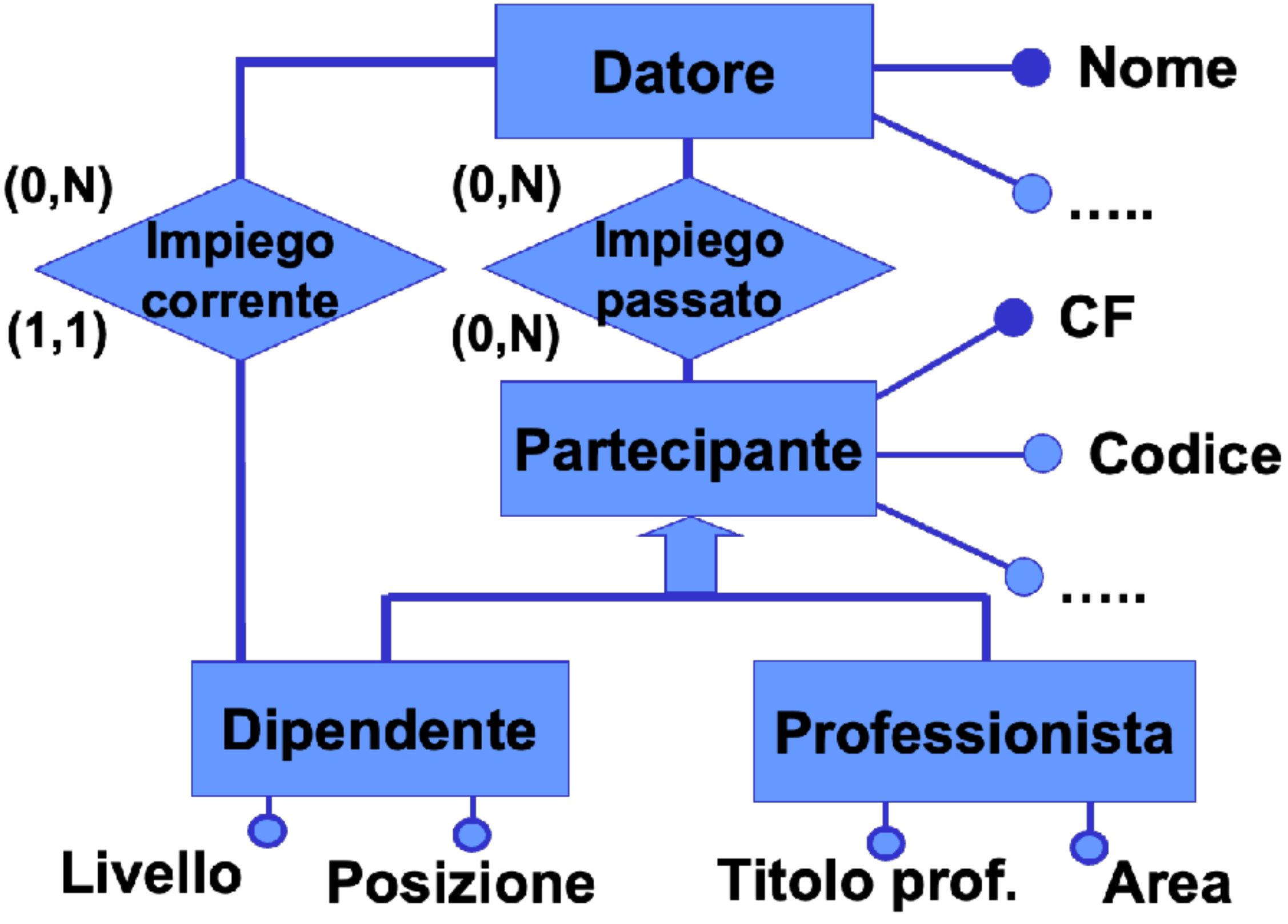
Esempio

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

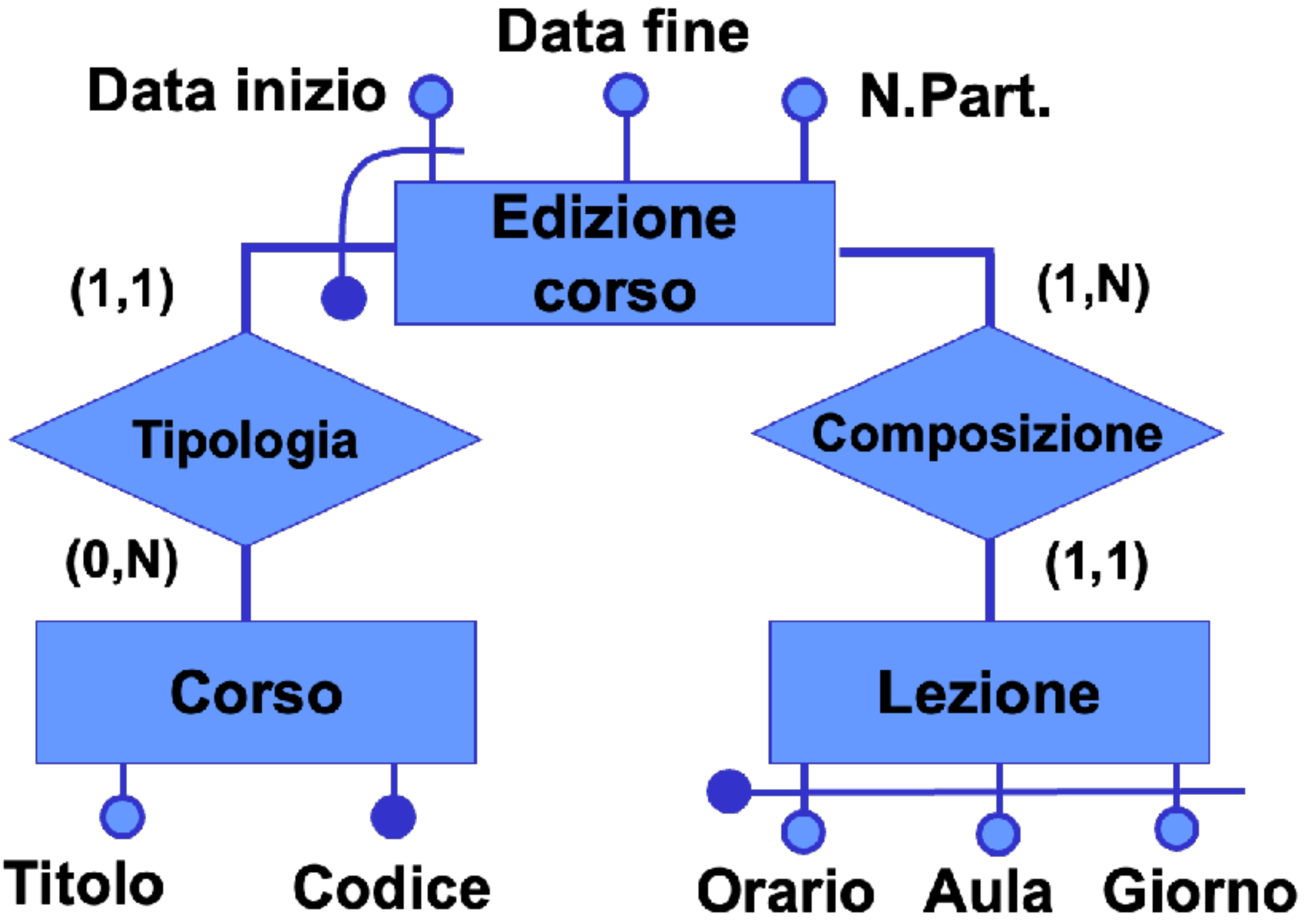
Frasi relative ai datori di lavoro
Relativamente ai datori di lavoro presenti e passati dei partecipanti, rappresentiamo il nome, l'indirizzo e il numero di telefono.

Frasi 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.


Datore
Nome
.....
(0,N)
(0,N)
Impiego corrente
Impiego passato
(1,1)
(0,N)
CF
Partecipante
Codice
.....
Dipendente
Professionista
Livello
Posizione
Titolo prof.
Area

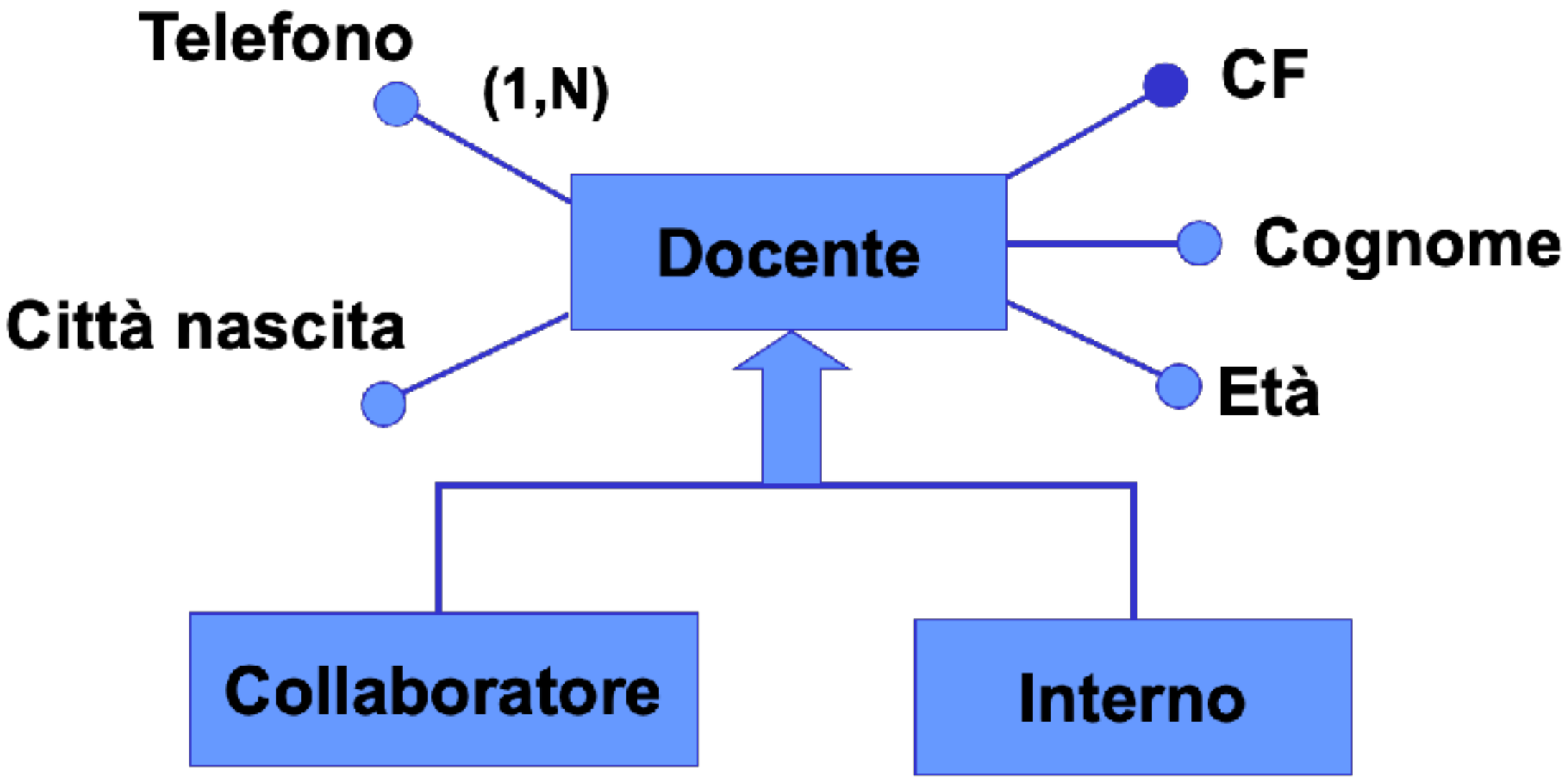
Esempio

Frasi 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.

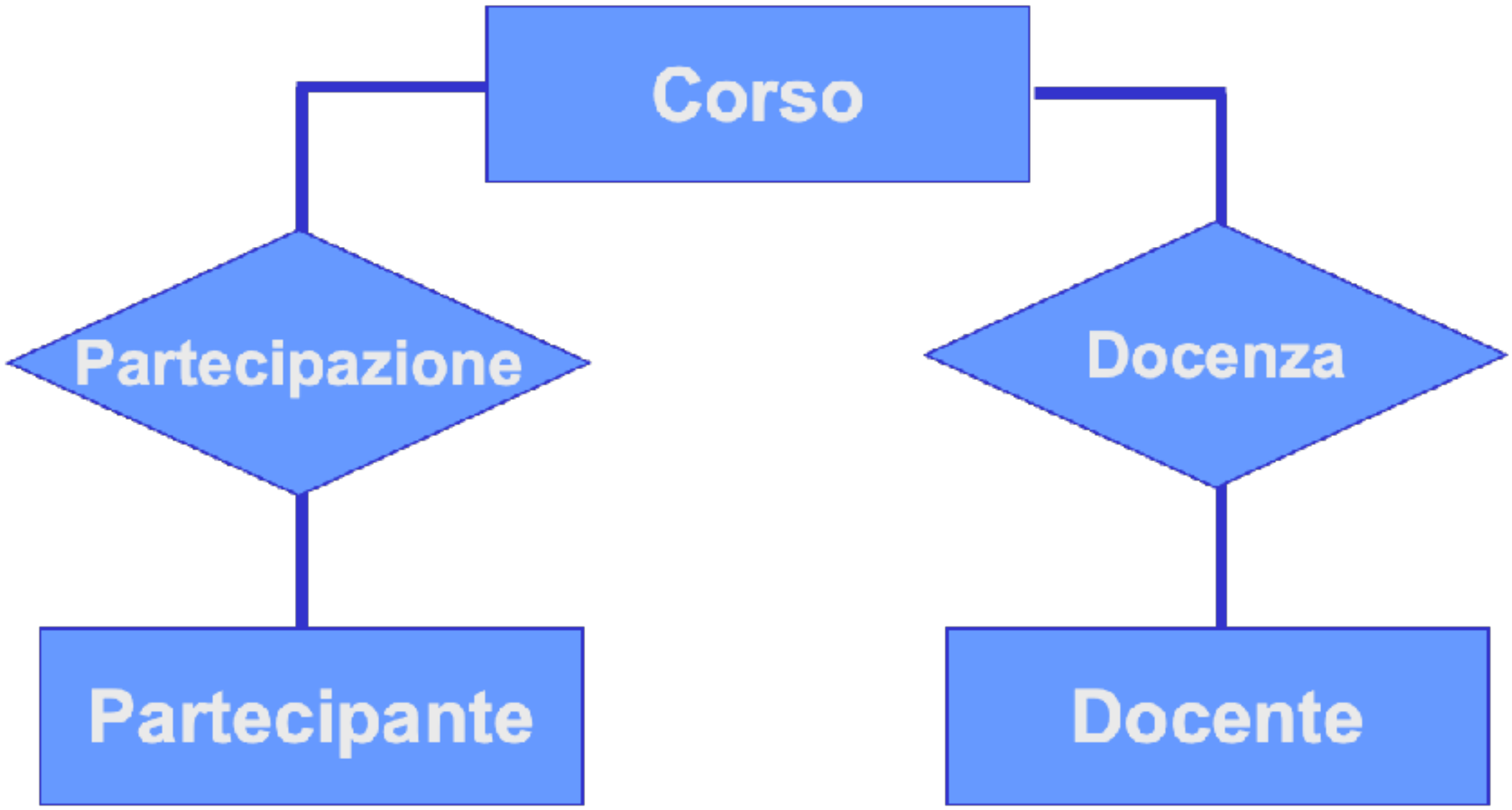

Data fine
Data inizio
N.Part.
Edizione corso
(1,1)
(1,N)
Tipologia
Composizione
(0,N)
(1,1)
Corso
Lezione
Titolo
Codice
Orario
Aula
Giorno

Esempio

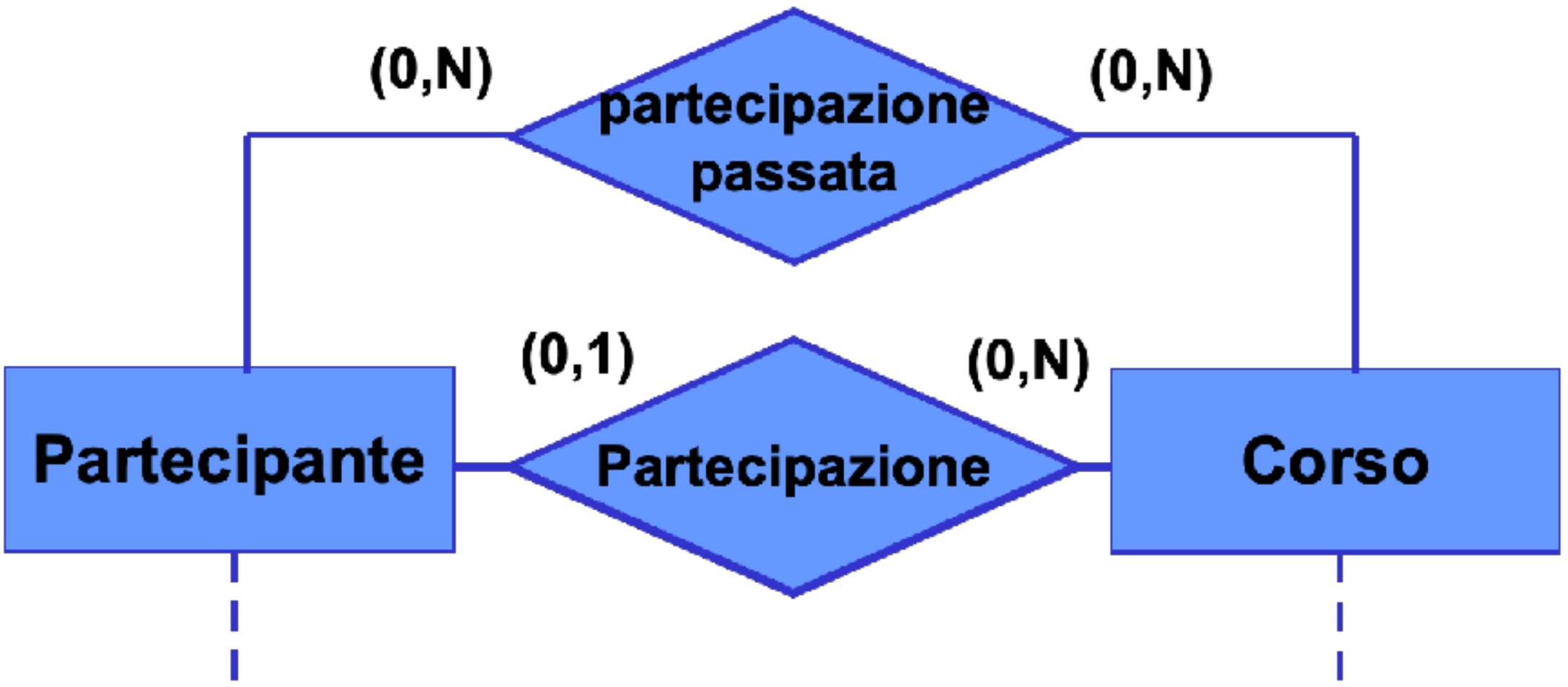
Frasi relative ai docenti
Per i docenti (circa 300), rappresentiamo il cognome, l'età, la città di nascita, tutti i numeri di telefono, il titolo del corso che insegnano, di quelli che hanno insegnato nel passato e di quelli che possono insegnare. I docenti possono essere dipendenti interni della società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


Telefono
(1,N)
CF
Docente
Cognome
Città nascita
Età
Collaboratore
Interno

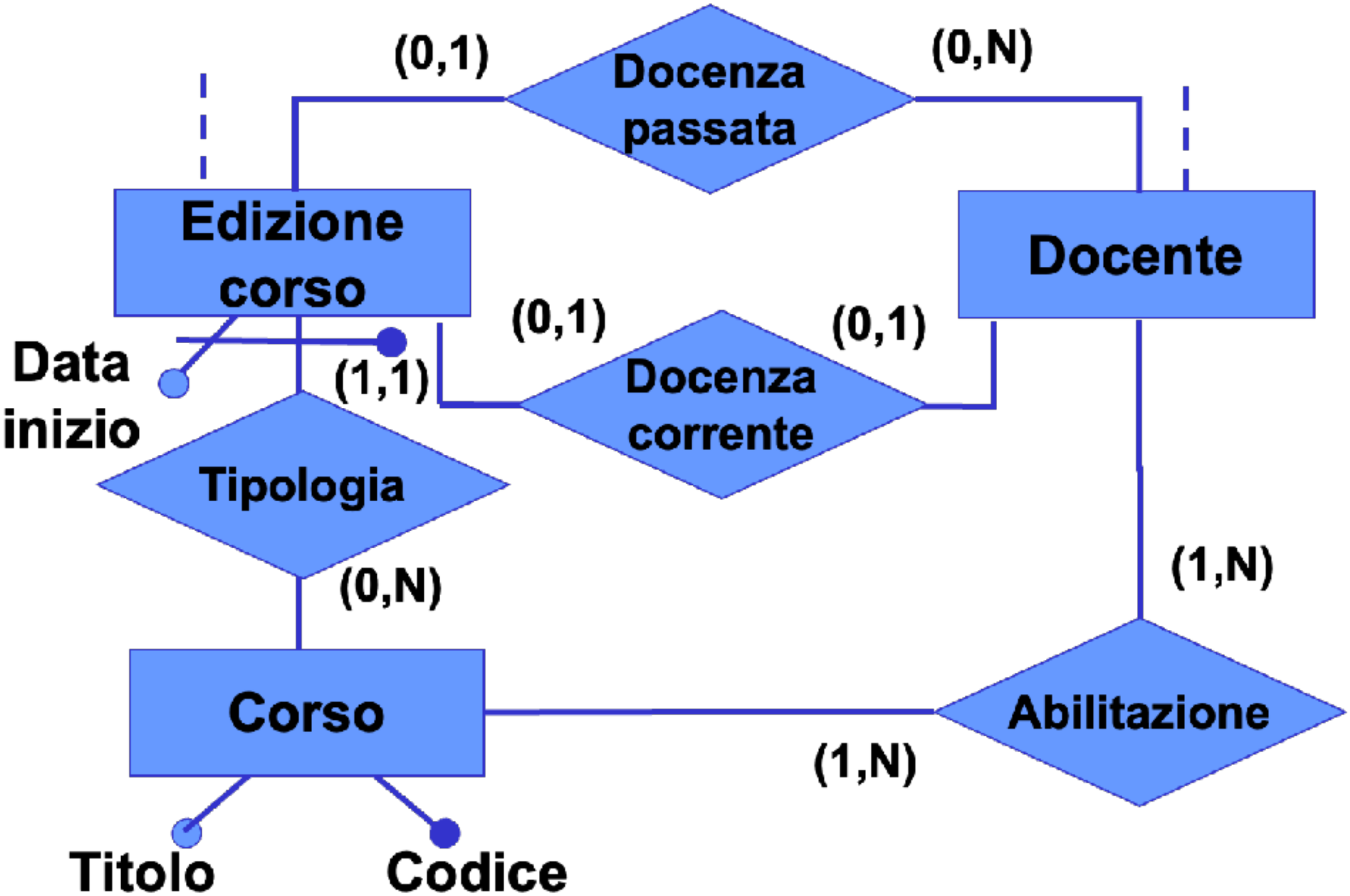
Integrazione



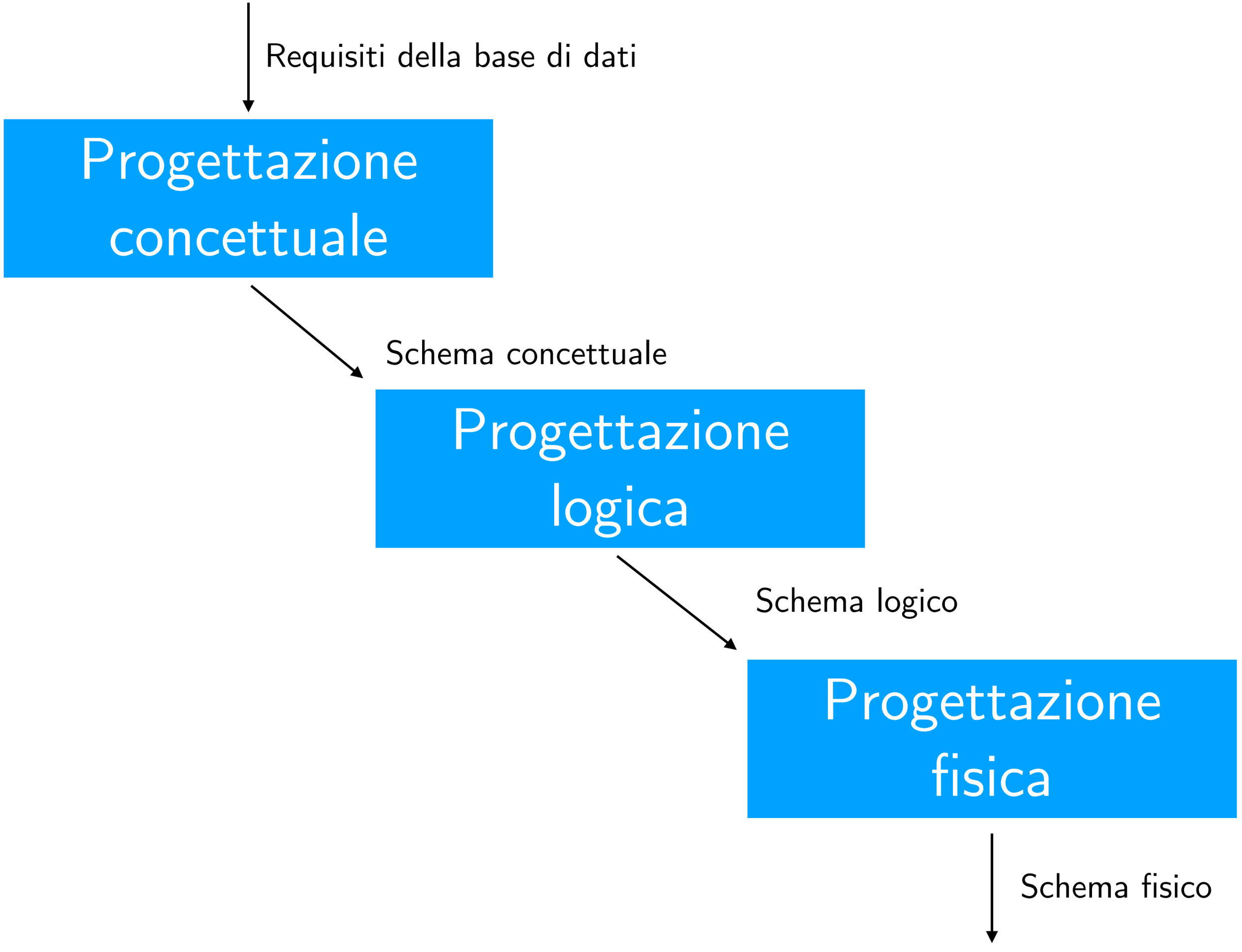
Integrazione



Integrazione

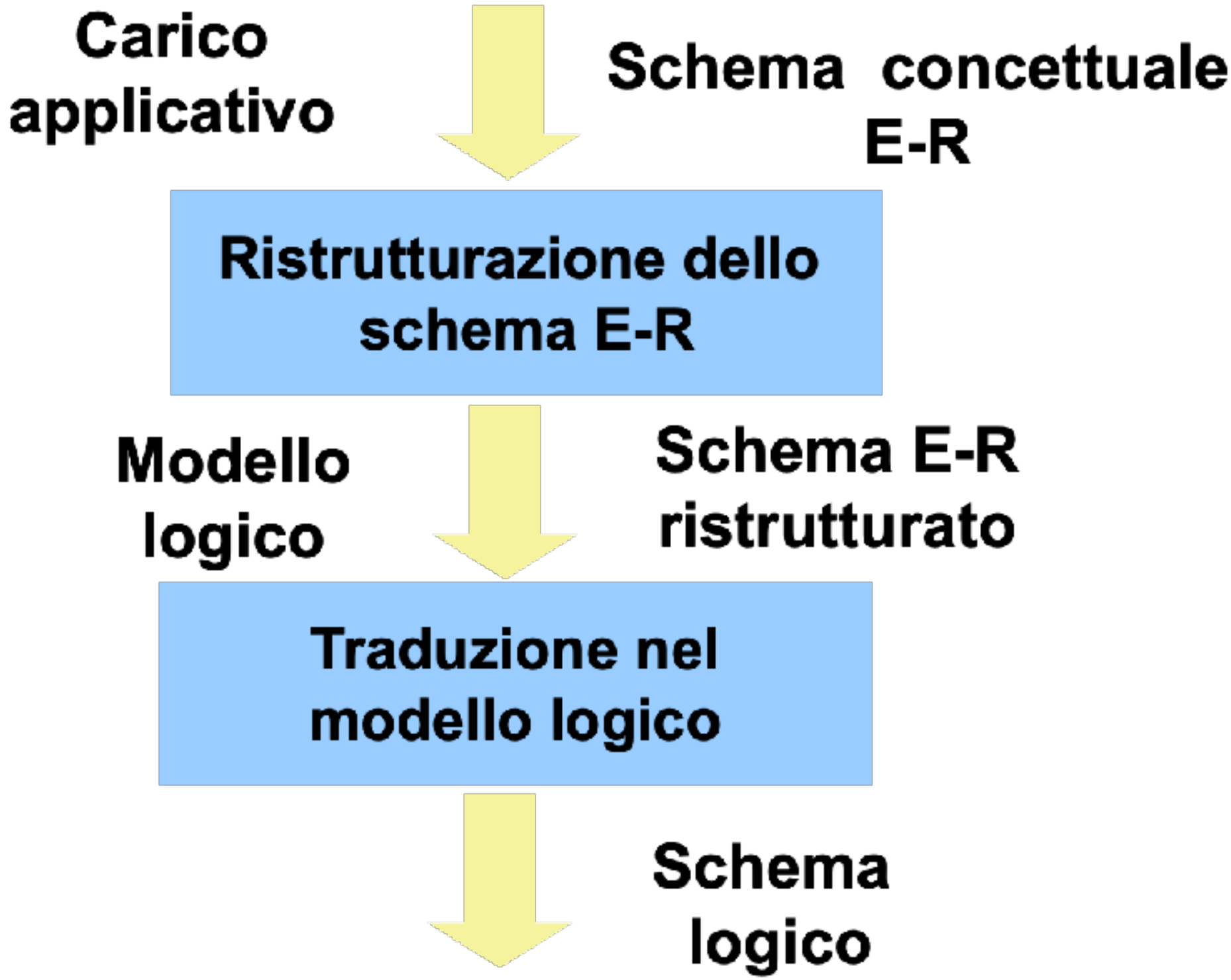


Progettazione Logica


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

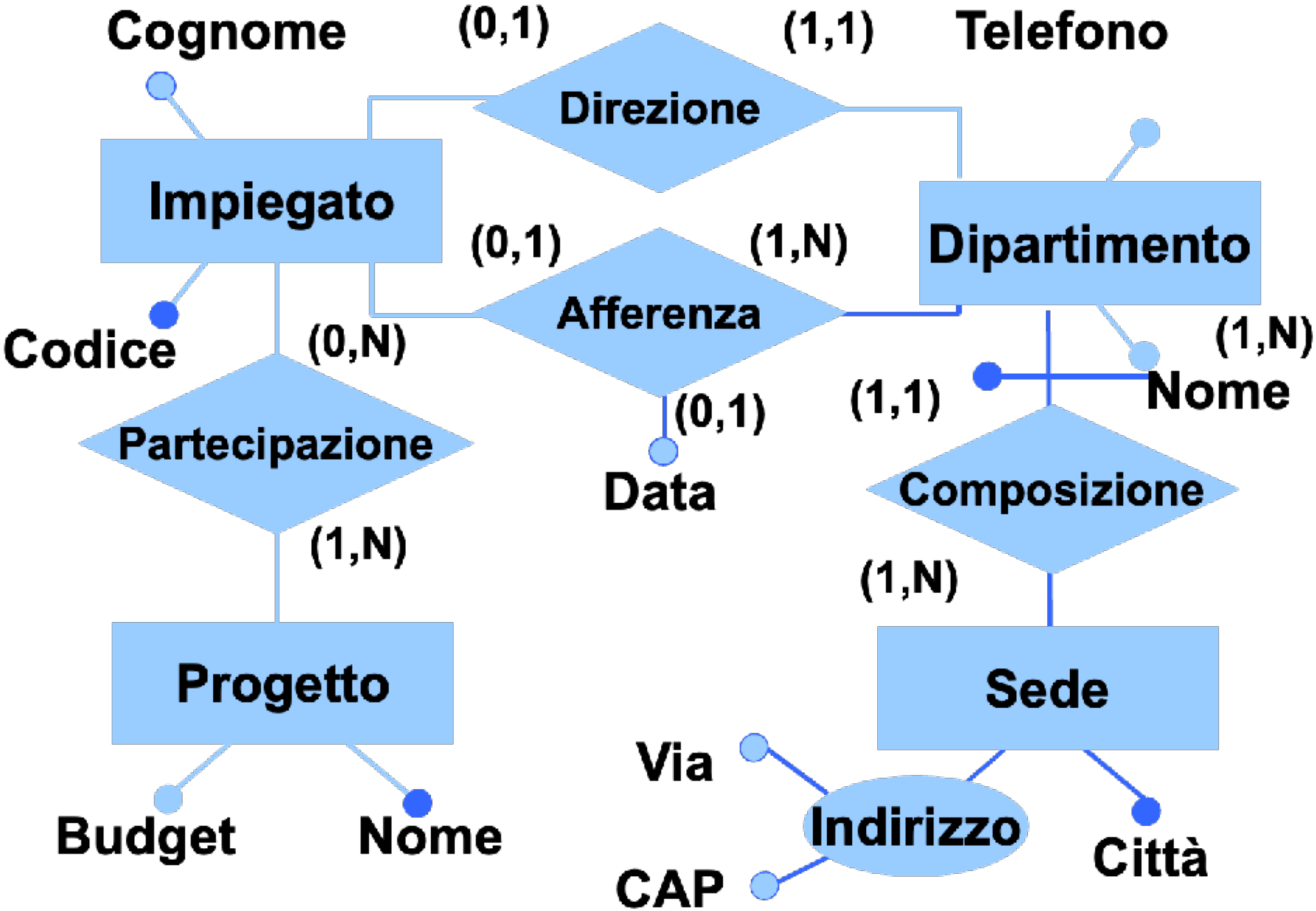

Cognome
(0,1)
(1,1)
Telefono
Direzione
Impiegato
Dipartimento
(0,1)
(1,N)
Afferenza
Codice
(0,N)
(1,N)
Nome
(1,1)
(0,1)
Data
Composizione
Partecipazione
(1,N)
(1,N)
Progetto
Sede
Via
Indirizzo
Città
CAP
Budget
Nome

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

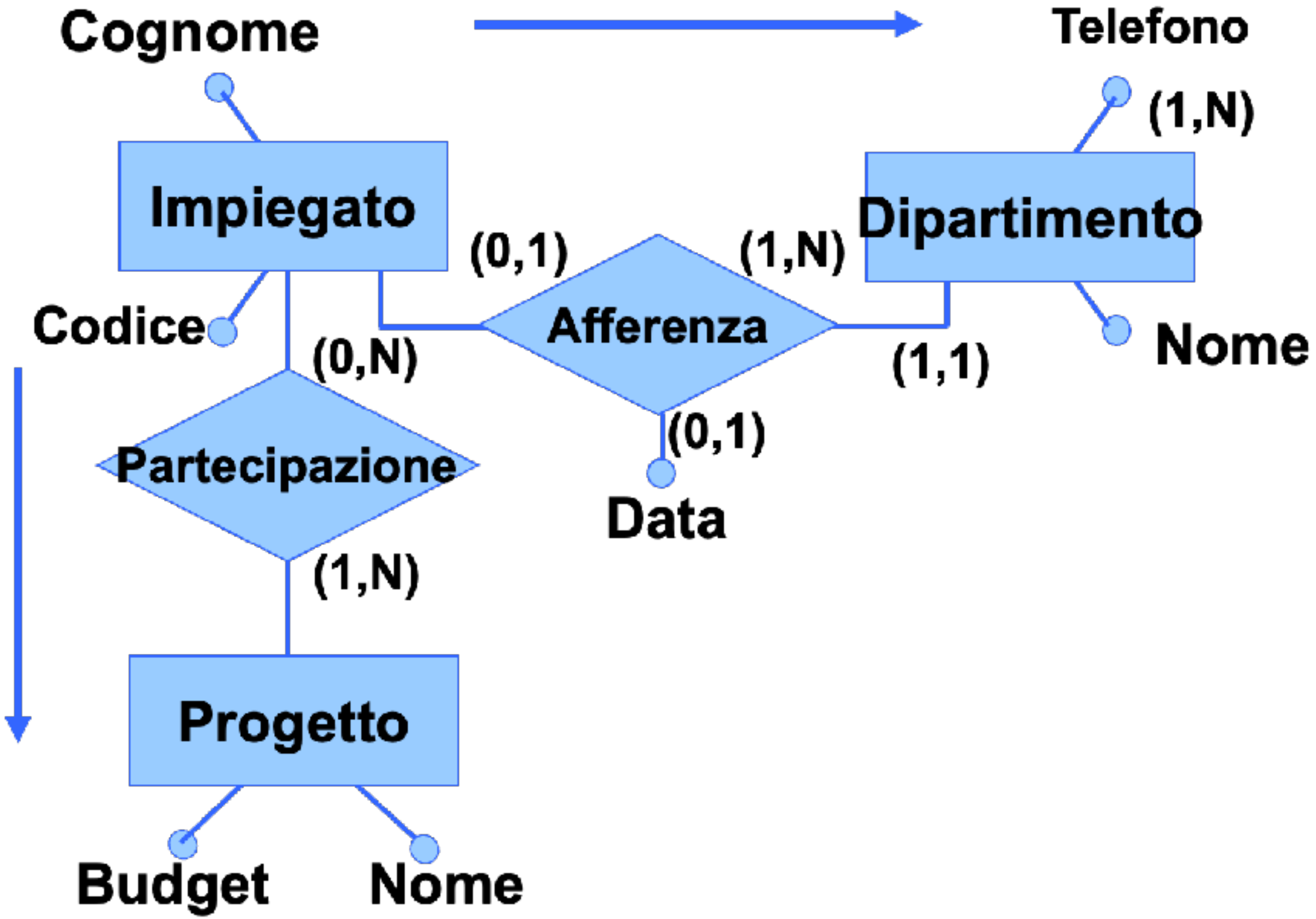


Tavola degli accessi

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

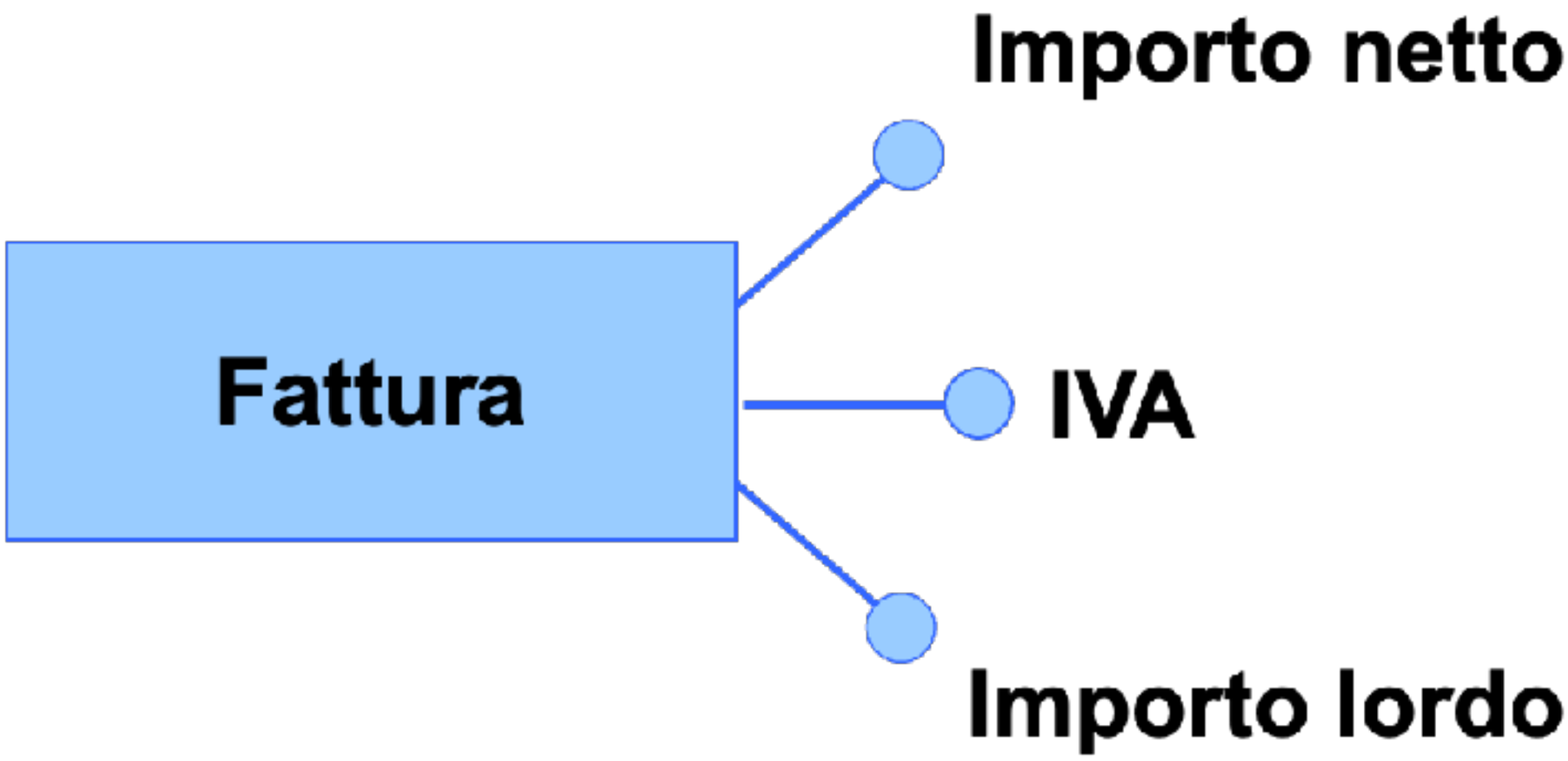
Analisi delle ridondanze

- 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**

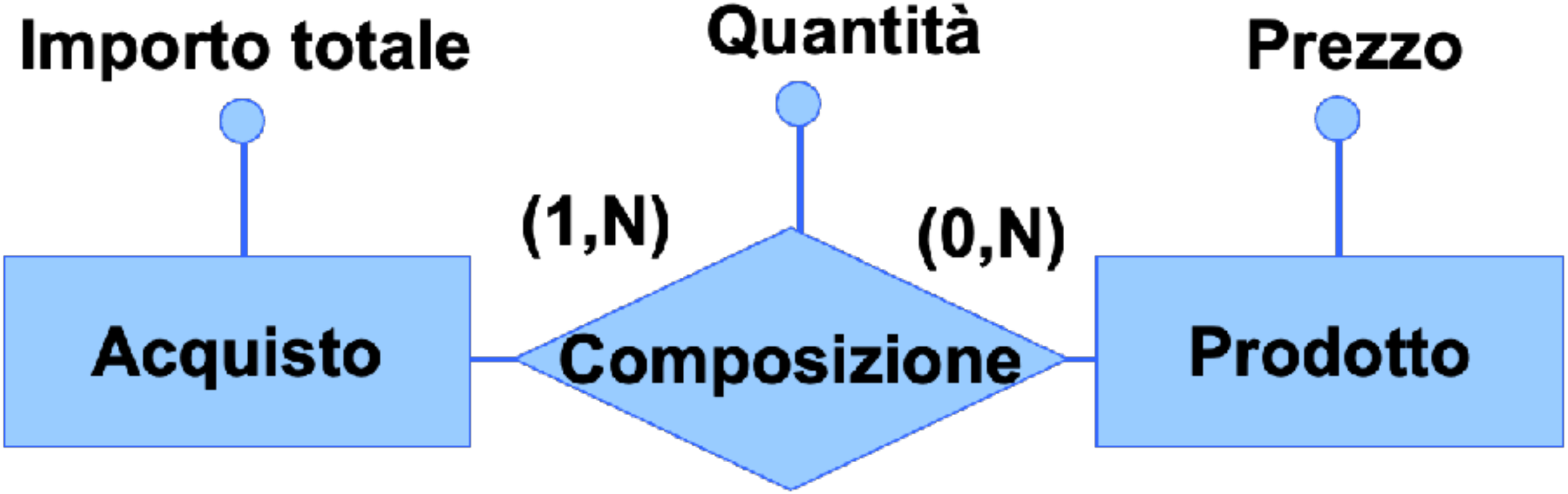
Forme di ridondanza in uno schema E-R

- **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*)

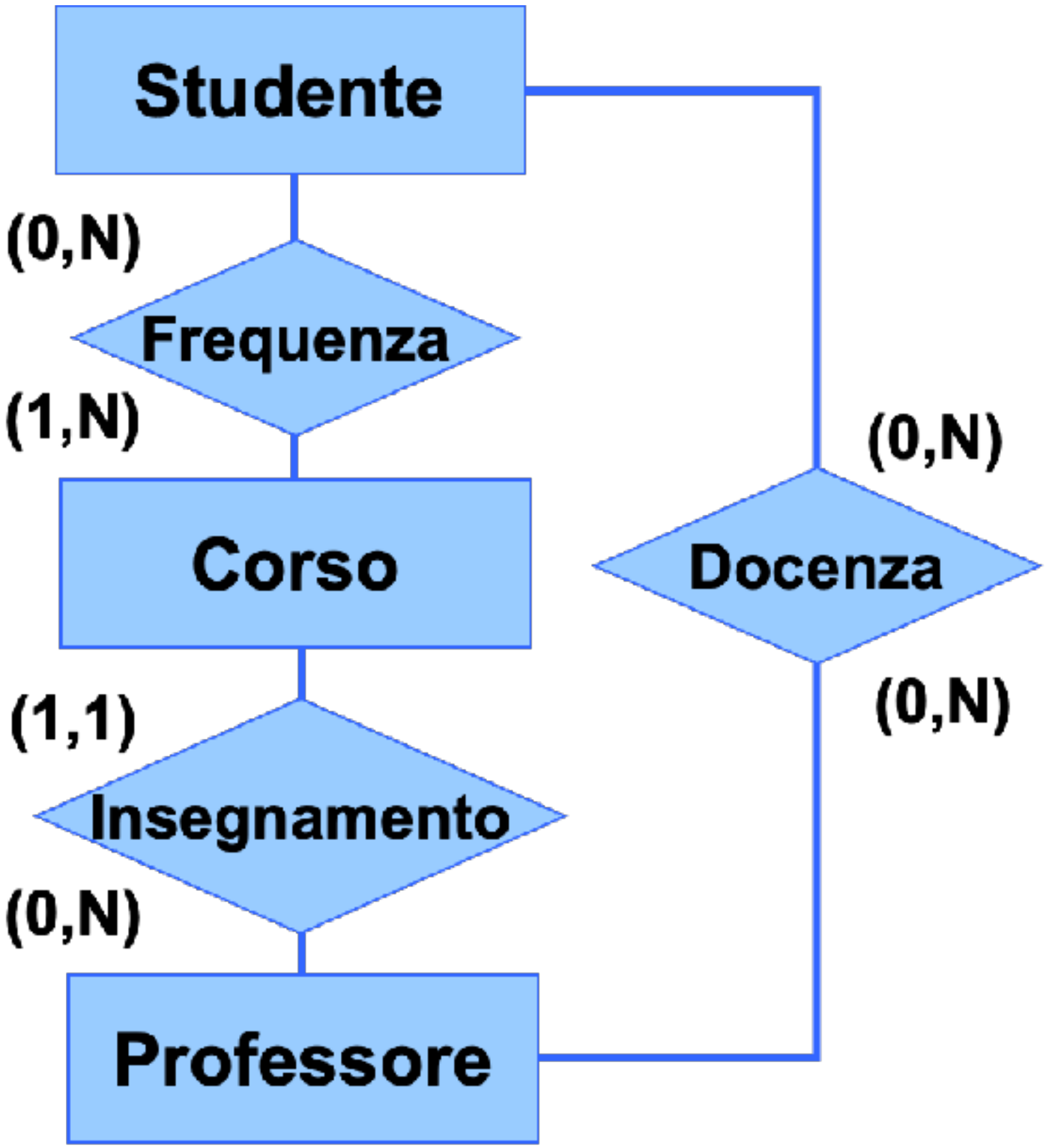
Attributo derivabile dalla stessa entità



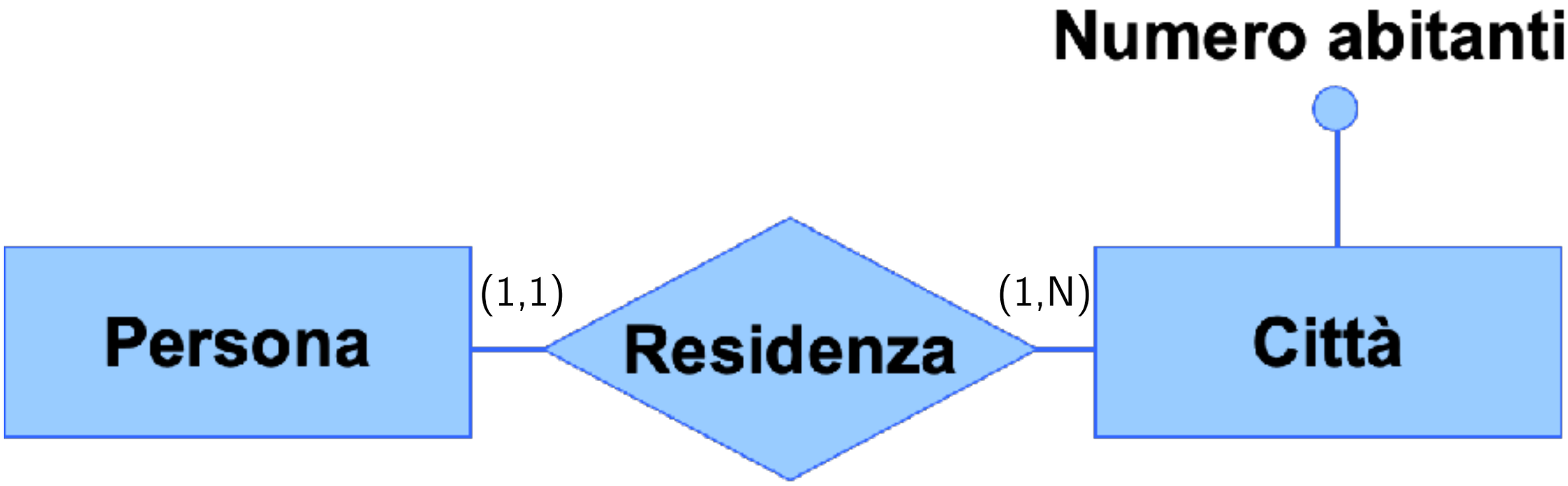
Attributo derivabile da altra entità



Ridondanza dovuta a ciclo

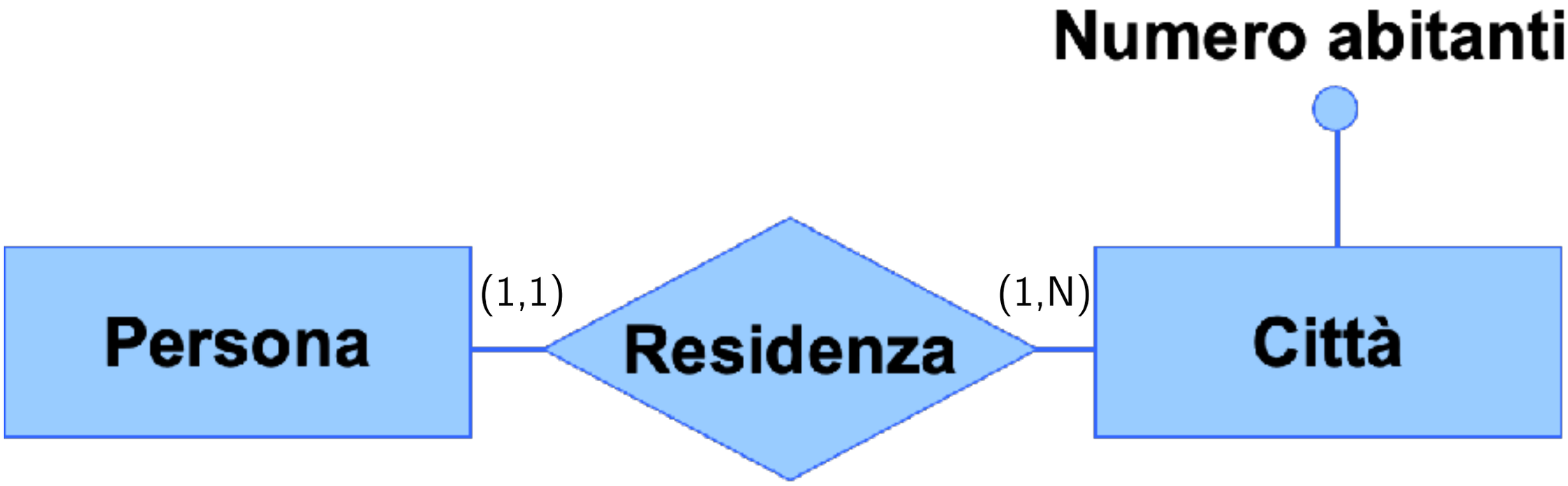


Analisi di una ridondanza



- L'attributo **Numero abitanti** è ridondante
- Per ottenerlo basta leggere e contare il numero di occorrenze di **Residenza** con una specifica città

Analisi di una ridondanza



- Abbiamo 200 città $\rightarrow$ 200 occorrenze di **Città**
- In ogni città abitano in media $5'000$ persone $\rightarrow$ $200 \times 5'000 = 1'000'000$ occorrenze di **Persona**
- Ogni occorrenza di **Persona** è in relazione con una e una sola occorrenza di **Città** $\rightarrow$ $1'000'000$ occorrenze di **Residenza**

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)

Tavola dei volumi e operazioni

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

- **Operazione 1**:
  - Scrivere una nuova occorrenza in **Persona**
  - Leggere un'occorrenza in **Città**, per conoscere **Numero abitanti**, incrementarlo di uno, e scrivere l'occorrenza in **Città** col nuovo valore dell'attributo
  - Scrivere una nuova occorrenza in **Residenza**

Tavola dei volumi e operazioni

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

- **Operazione 2**:
  - Leggere un'occorrenza in **Città**

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

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

- **Operazione 1**:
  - Scrivere una nuova occorrenza in **Persona**
  - Scrivere una nuova occorrenza in **Residenza**

Assenza di ridondanza

Operazione 2

Concetto	Costrutto	Accessi	Tipo
Città	Entità	1	L
Residenza	Relazione	5000	L

- **Operazione 2**:
  - Leggere un'occorrenza in **Città**
  - Leggere circa $5'000$ occorrenze in **Residenza**

Assenza di ridondanza

Operazione 1

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

Operazione 2

Concetto	Costrutto	Accessi	Tipo
Città	Entità	1	L
Residenza	Relazione	5000	L

Costi

- **Presenza di ridondanza**:
  - Costi:
    - Operazione 1: $1'500$ accessi in scrittura e 500 accessi in lettura al giorno
    - Operazione 2: trascurabile (2)
  - Contiamo doppi gli accessi in scrittura
    - Totale di $1'500 \times 2 + 500 = 3'500$ accessi al giorno
- **Assenza di ridondanza**:
  - Costi:
    - Operazione 1: $1'000$ accessi in scrittura
    - Operazione 2: $10'000$ accessi in lettura al giorno
  - Contiamo doppi gli accessi in scrittura
    - Totale di $1'000 \times 2 + 10'000 = 12'000$ accessi al giorno

Attività di ristrutturazione

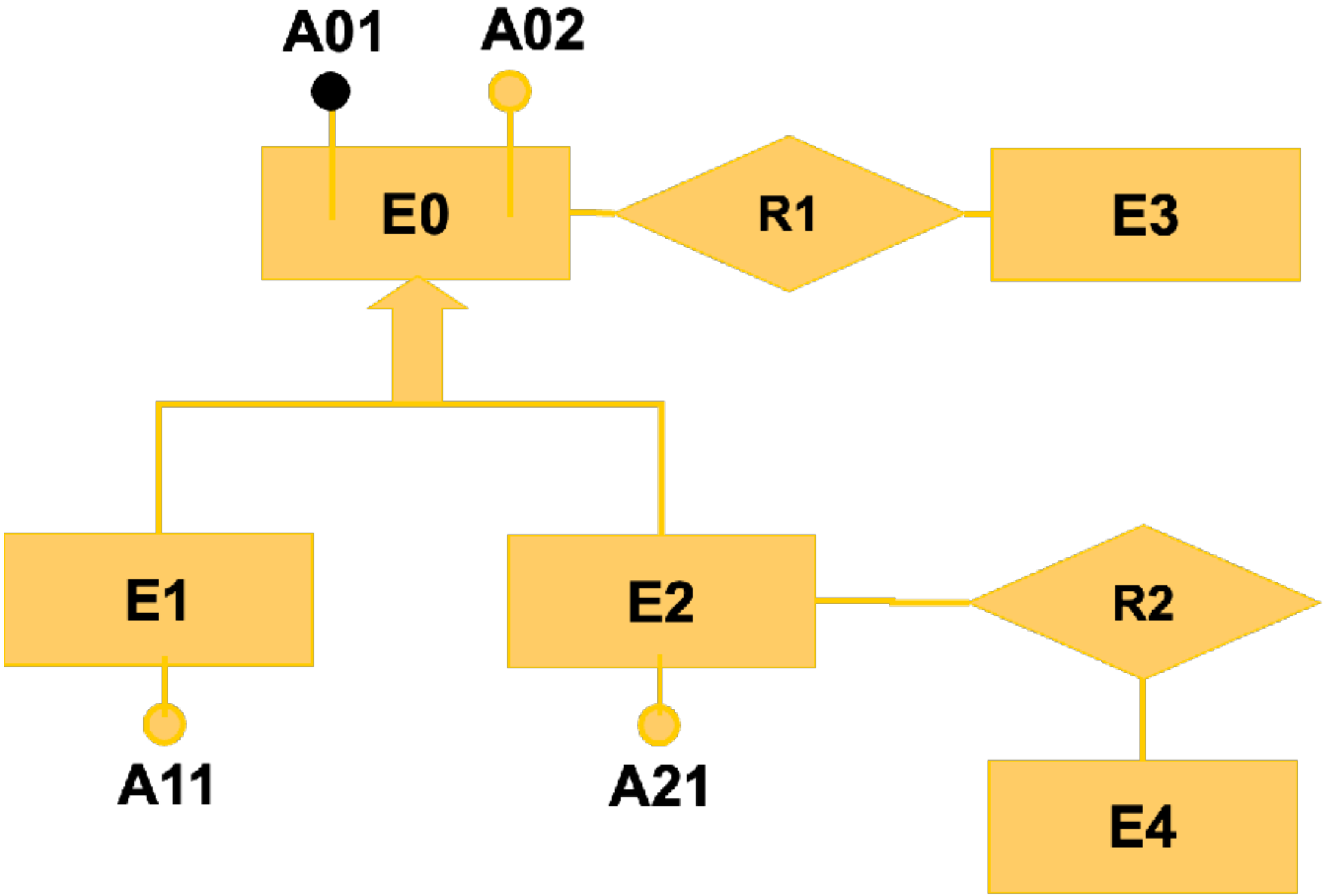
- Analisi delle ridondanze
- **Eliminazione delle generalizzazioni**
- Partizionamento/accorpamento di entità e *relationship*
- Scelta degli identificatori primari

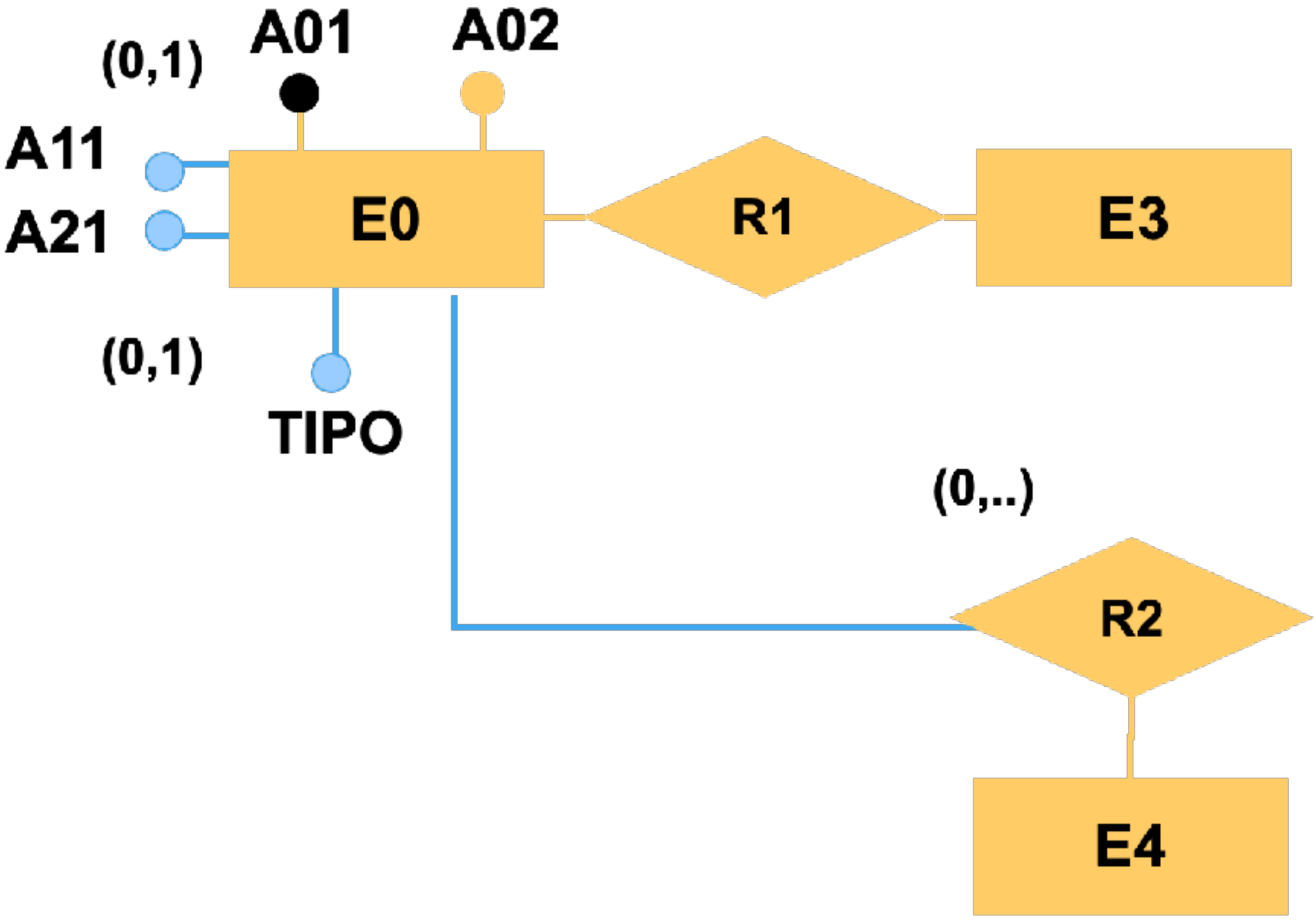
Le gerarchie nel modello relazionale

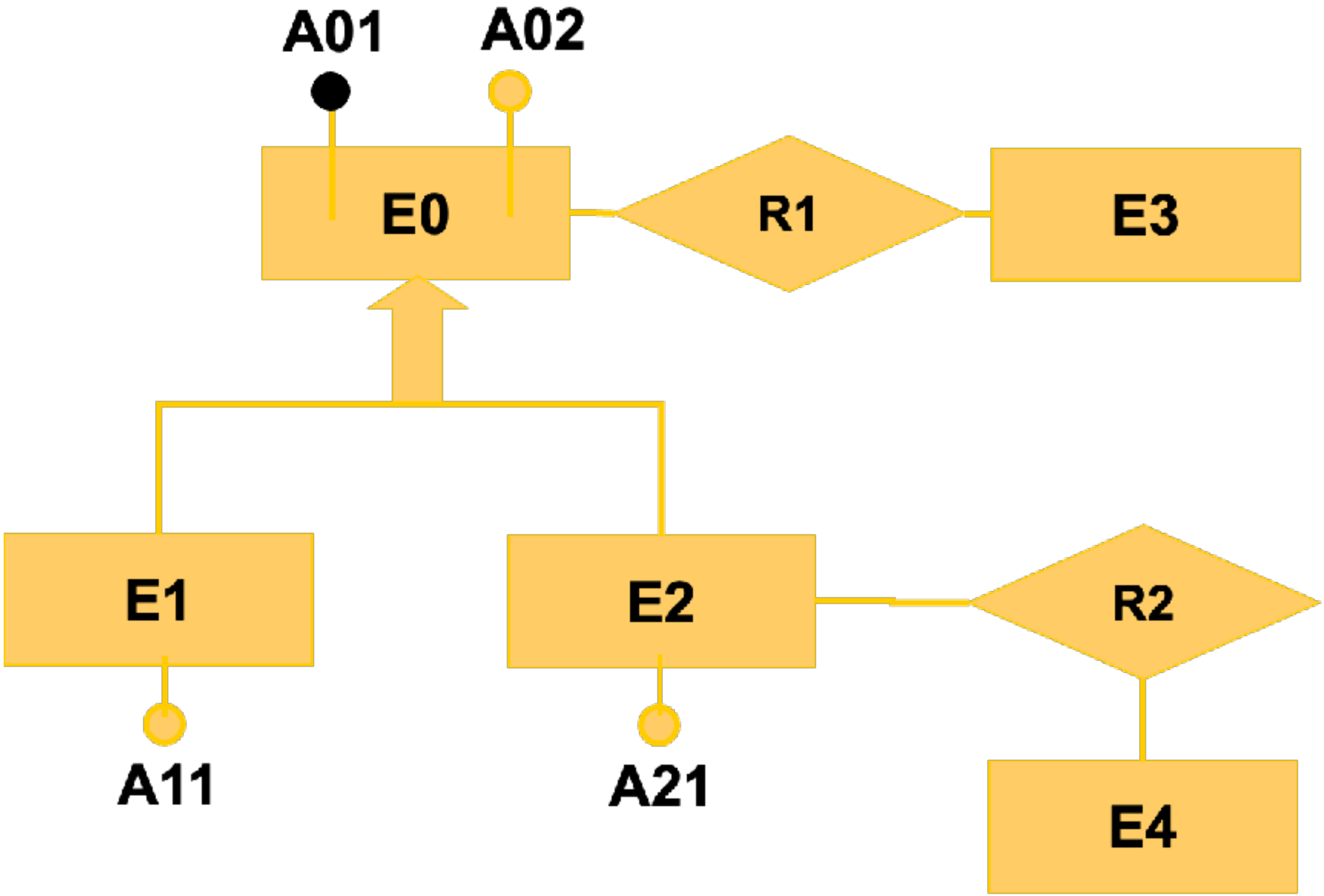
- Il modello relazionale **non può rappresentare** direttamente le generalizzazioni
- **Entità** e ***relationship*** sono invece direttamente rappresentabili
- Si **eliminano perciò le gerarchie**, sostituendole con entità e *relationship*

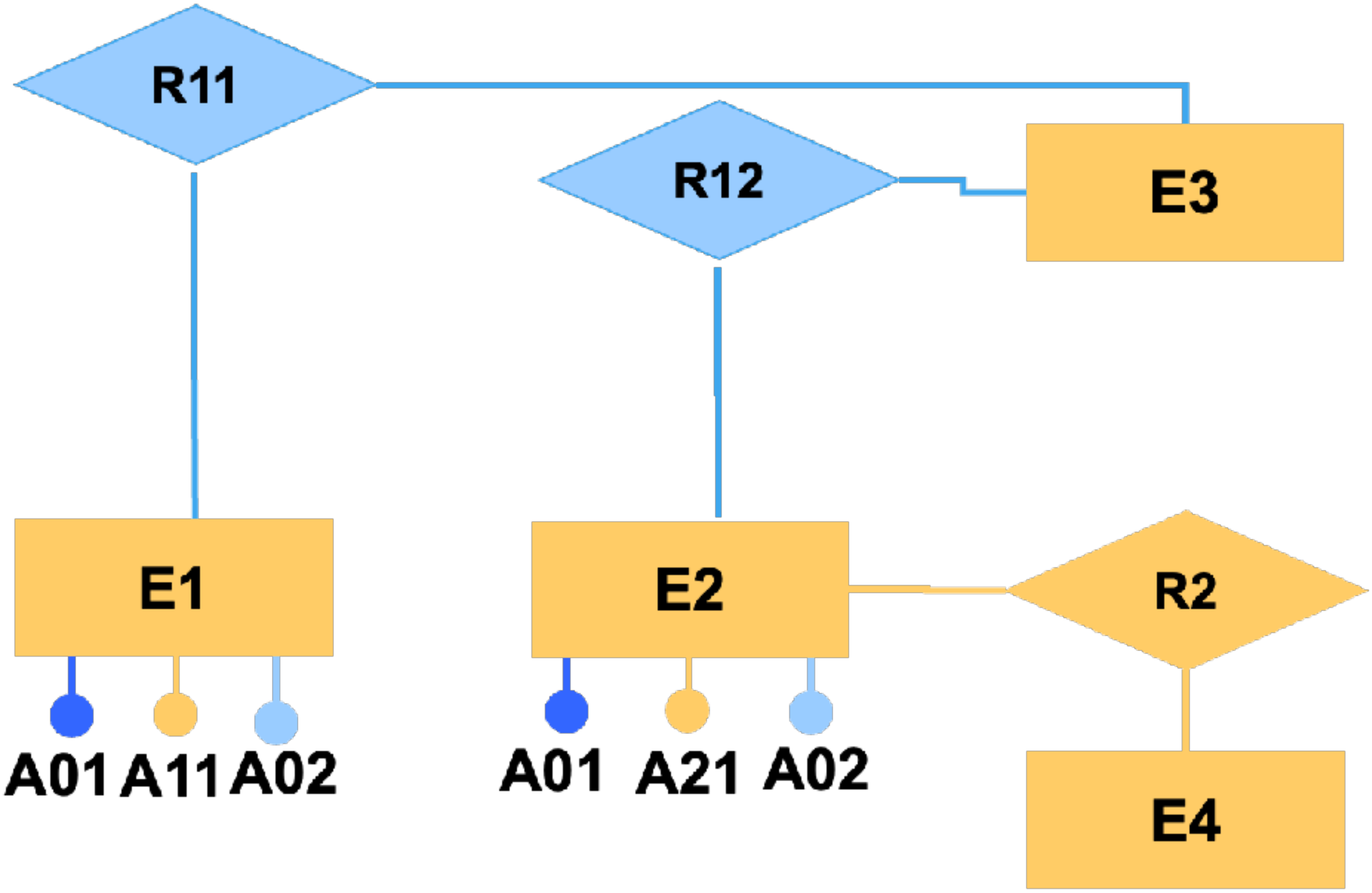
Possibilità

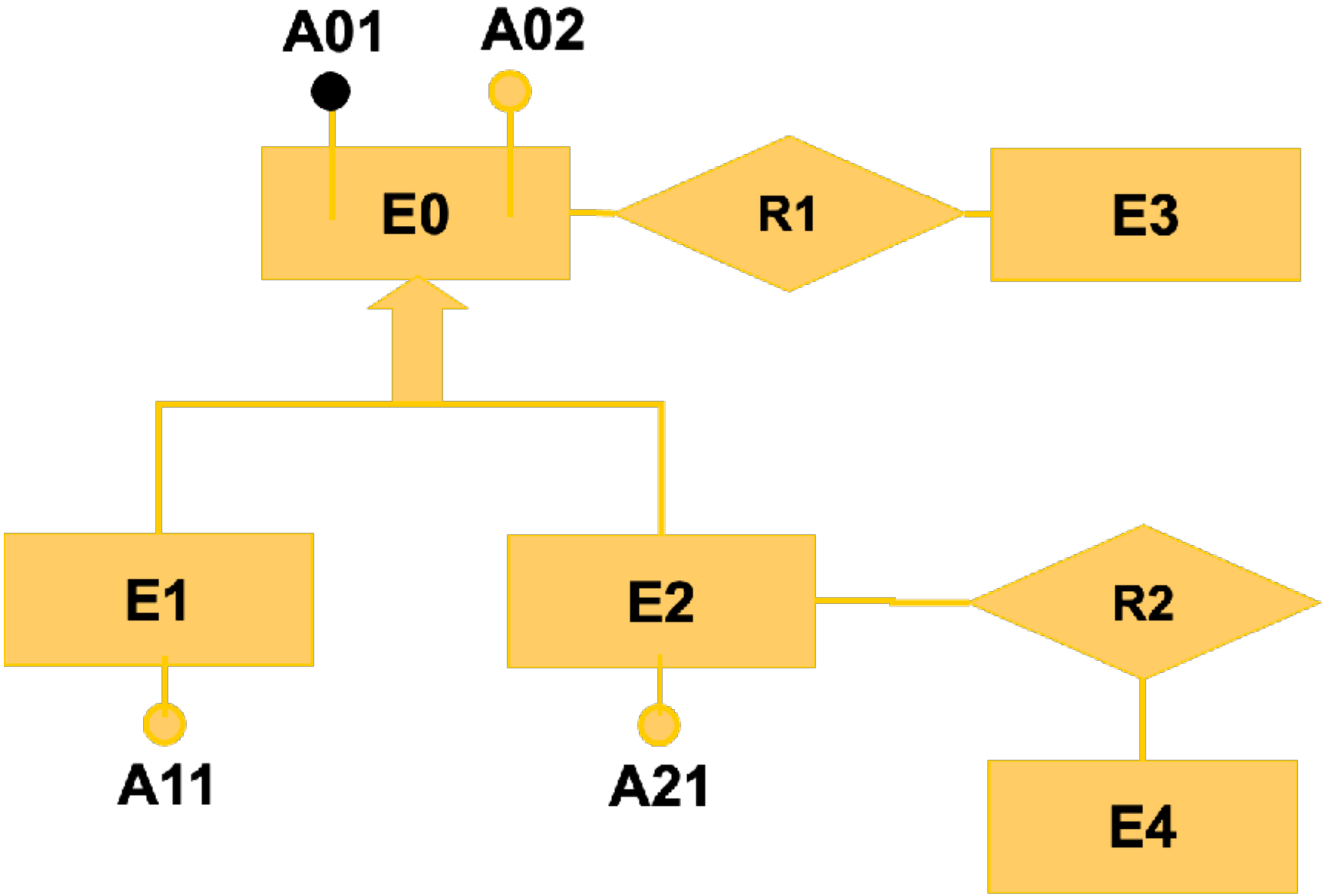
1. **Accorpamento delle figlie** della generalizzazione nel genitore
2. **Accorpamento del genitore** della generalizzazione nelle figlie
3. **Sostituzione** della generalizzazione con *relationship*

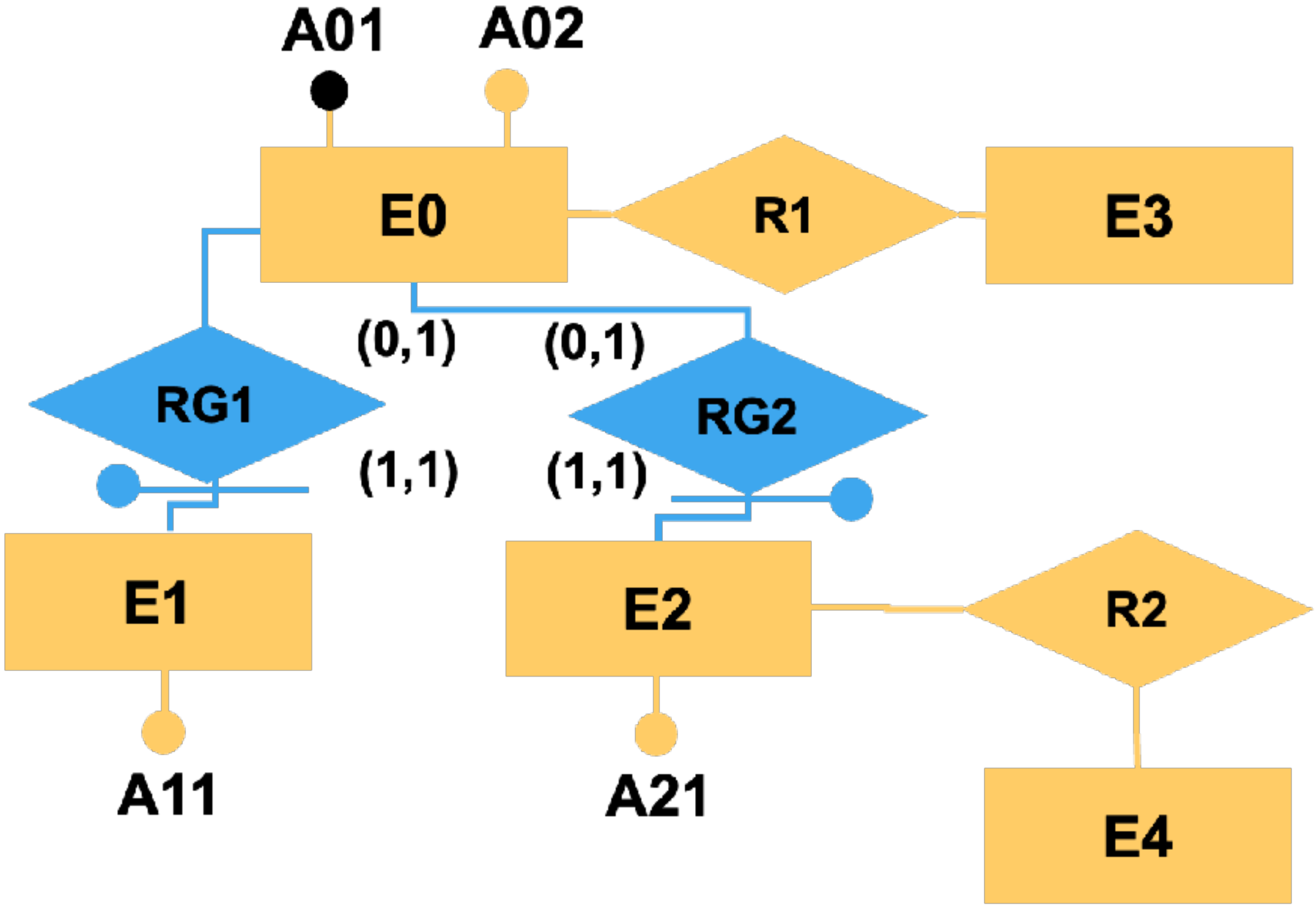

A01
A02
E0
R1
E3
E1
E2
R2
A11
A21
E4


A01
A02
(0,1)
A11
A21
E0
R1
E3
(0,1)
TIPO
(0,..)
R2
E4


A01
A02
E0
R1
E3
E1
E2
R2
A11
A21
E4

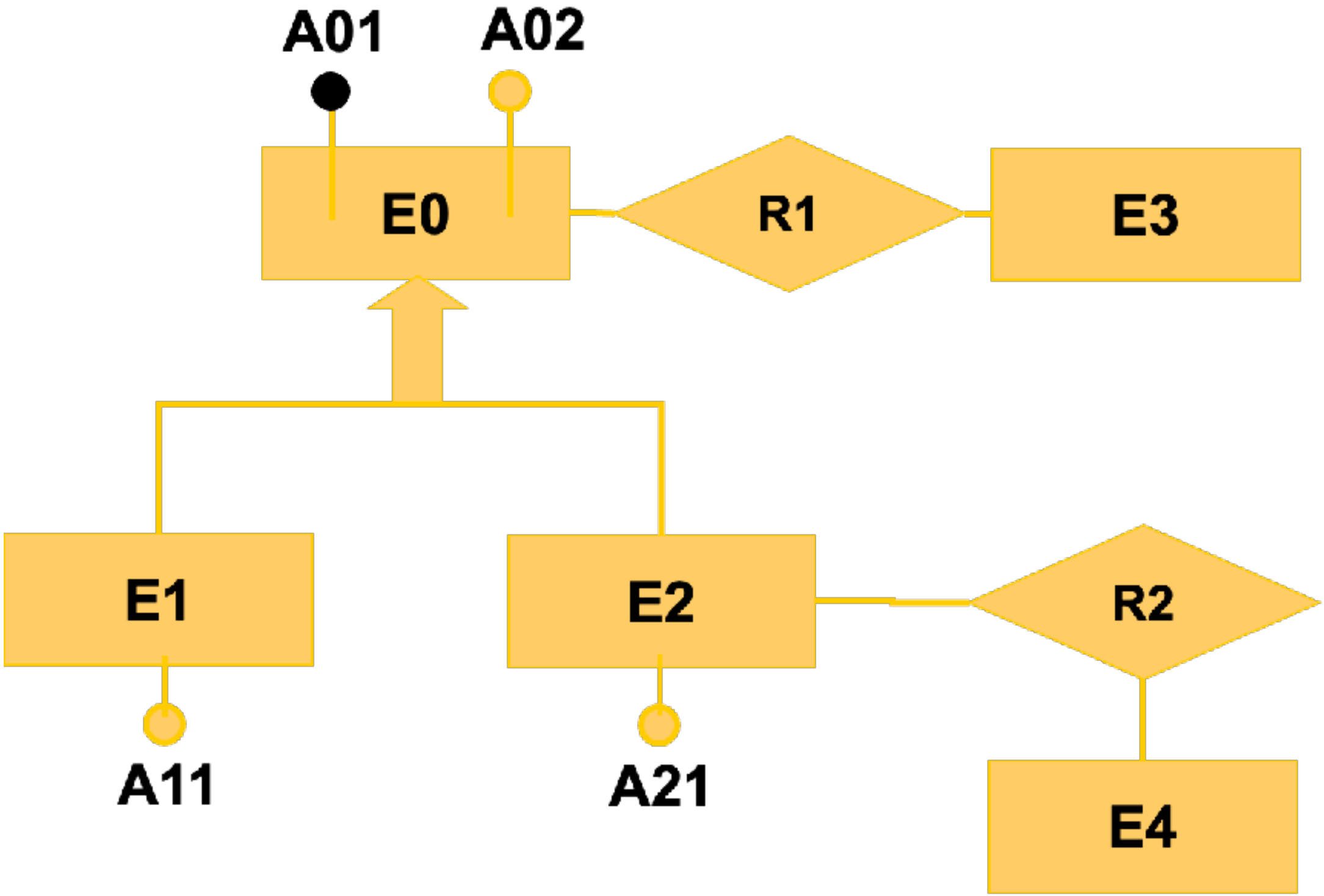

R11
R12
E3
E1
E2
R2
E4
A01
A11
A02
A01
A21
A02

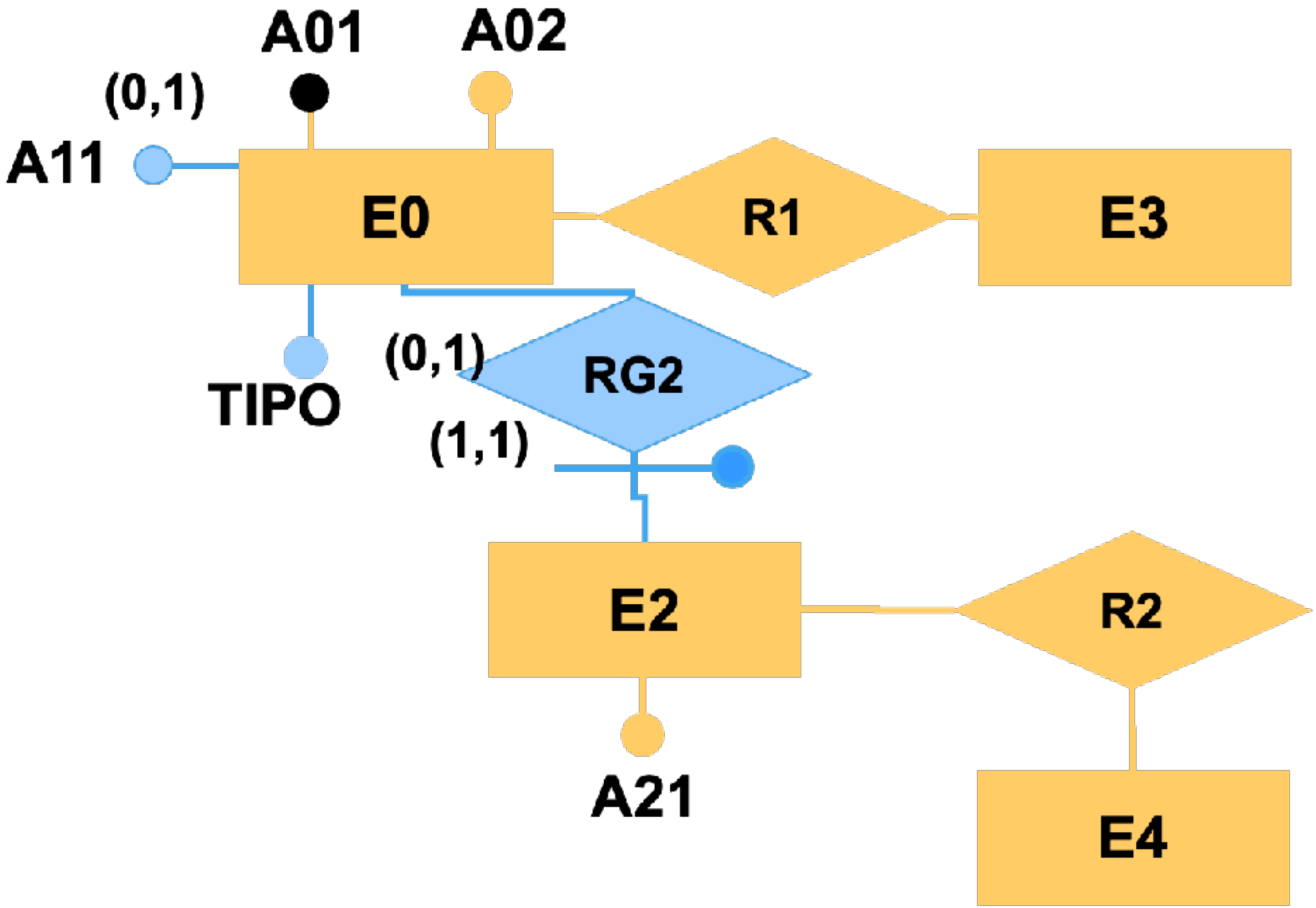

A01
A02
E0
R1
E3
E1
E2
R2
A11
A21
E4


A01
A02
E0
R1
E3
(0,1)
(0,1)
RG1
RG2
(1,1)
(1,1)
E1
E2
R2
A11
A21
E4

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.


A01
A02
E0
R1
E3
E1
E2
R2
A11
A21
E4


A01
A02
(0,1)
A11
E0
R1
E3
TIPO
(0,1)
RG2
(1,1)
E2
R2
A21
E4

Attività di ristrutturazione

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

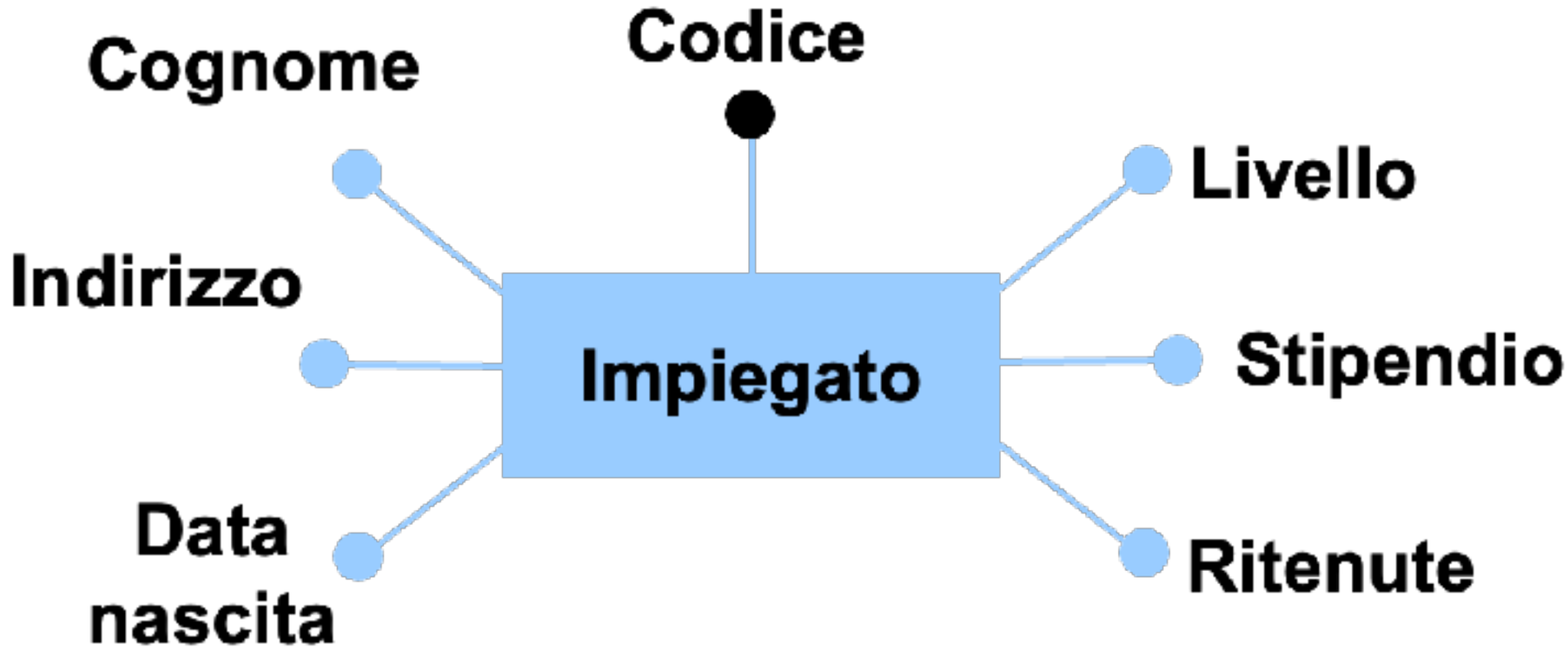
Motivazione

- 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**

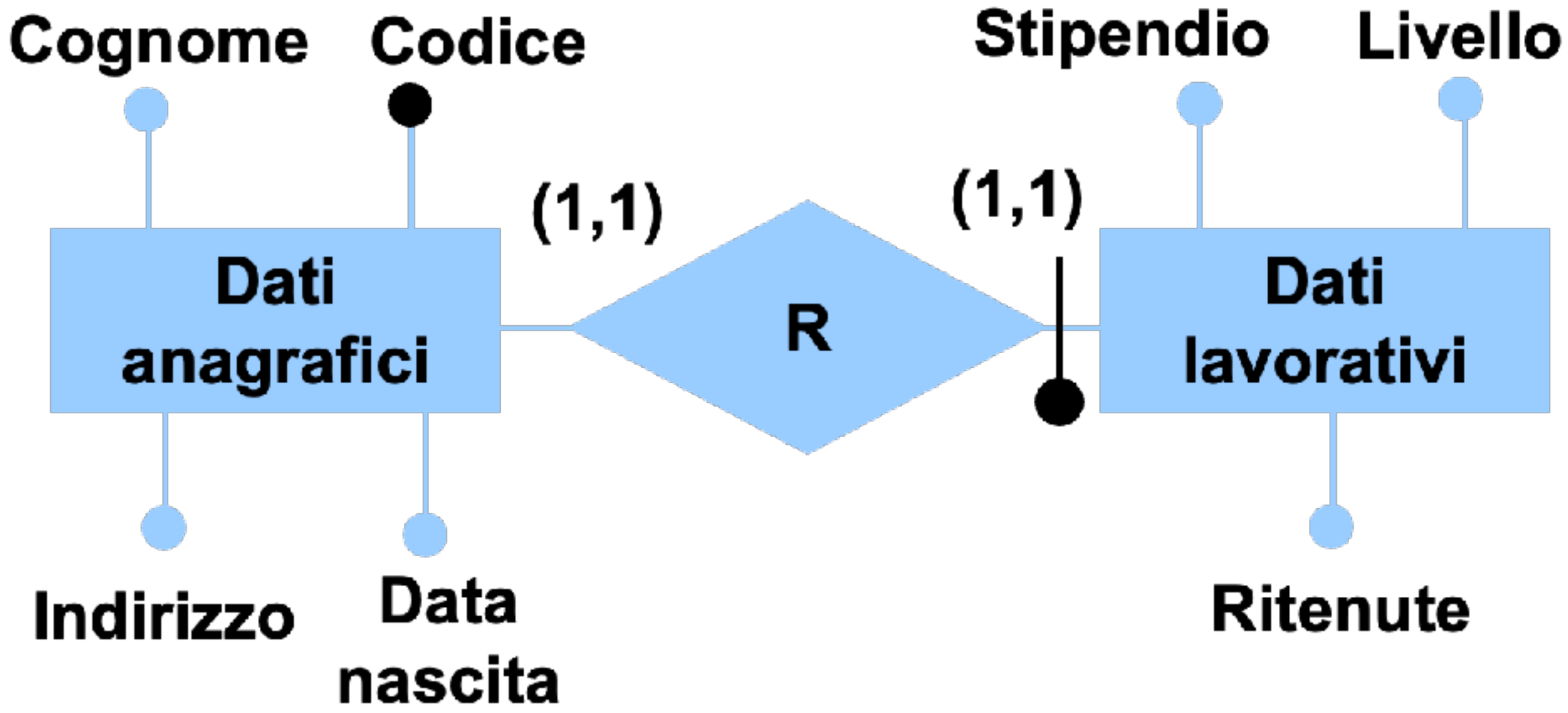
Casi principali

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

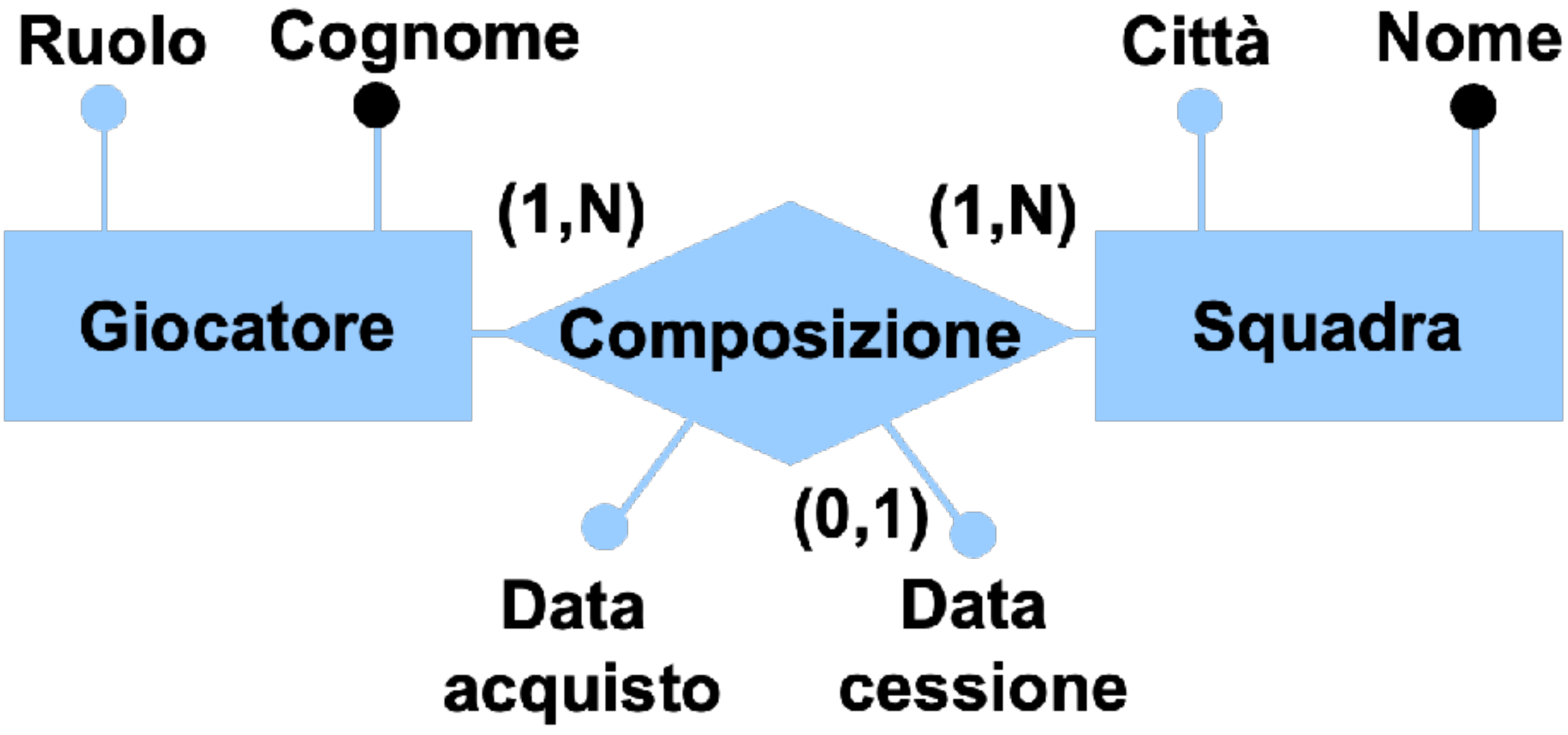
Partizionamento verticale di entità



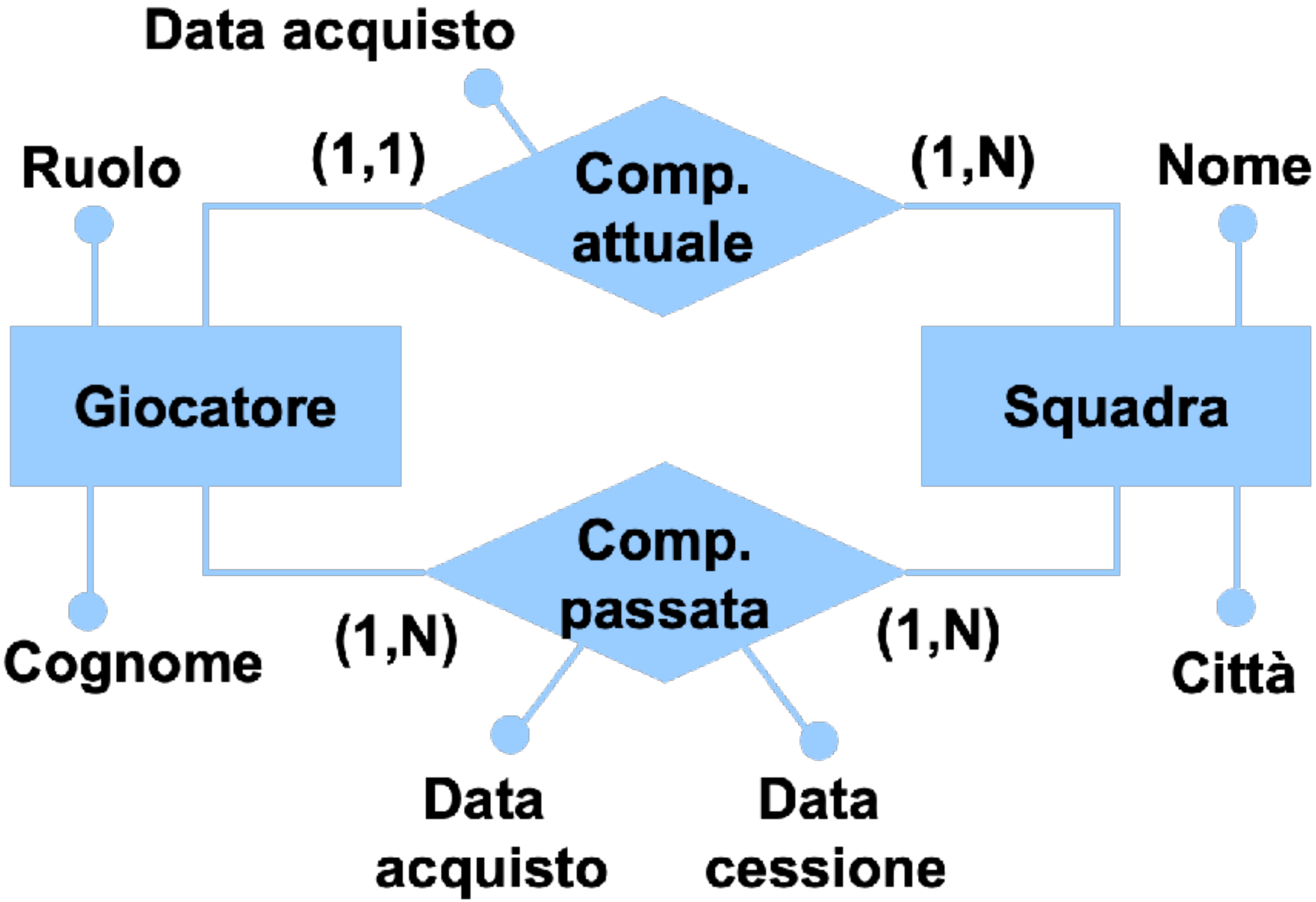
Partizionamento verticale di entità



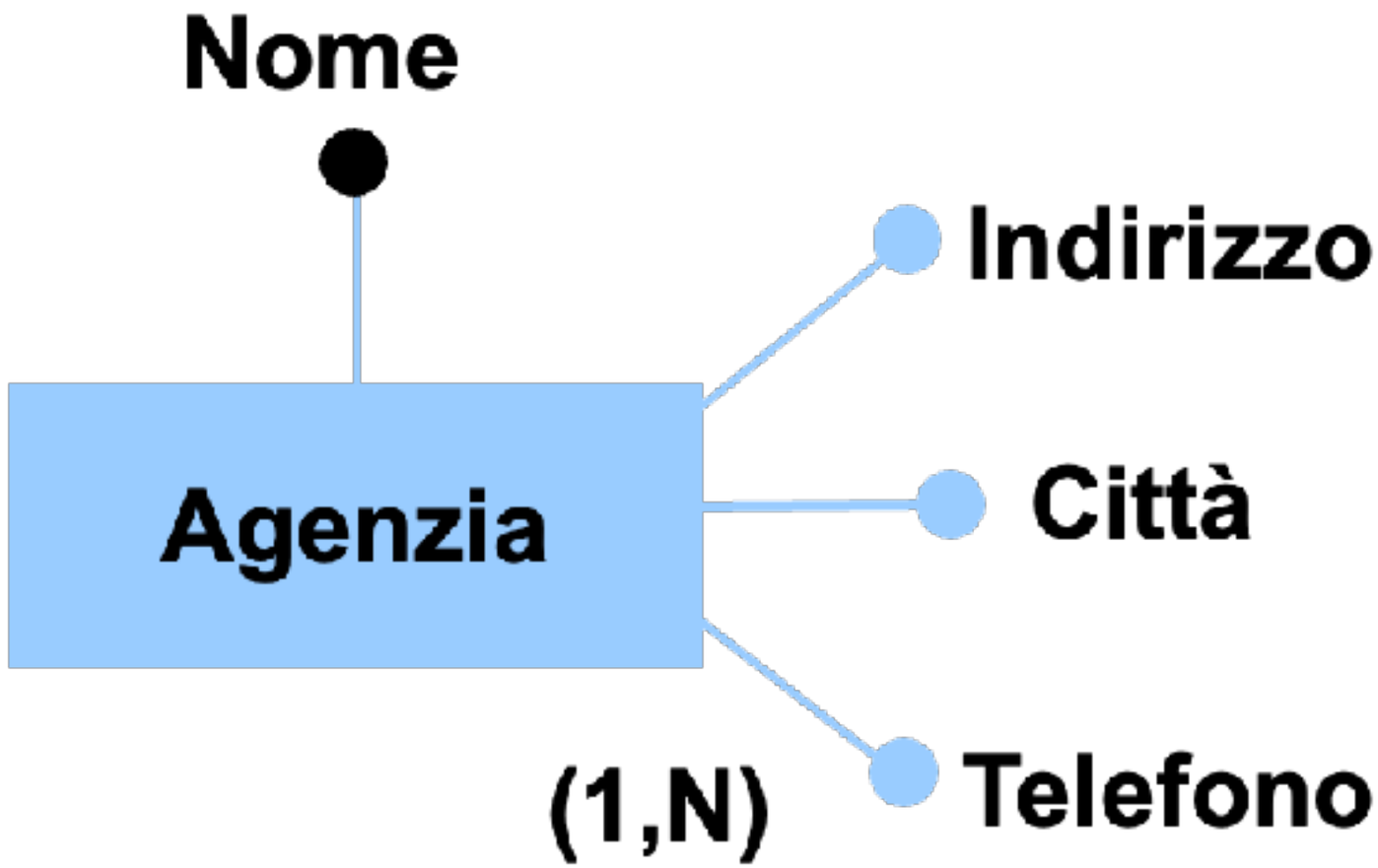
Partizione orizzontale di *relationship*



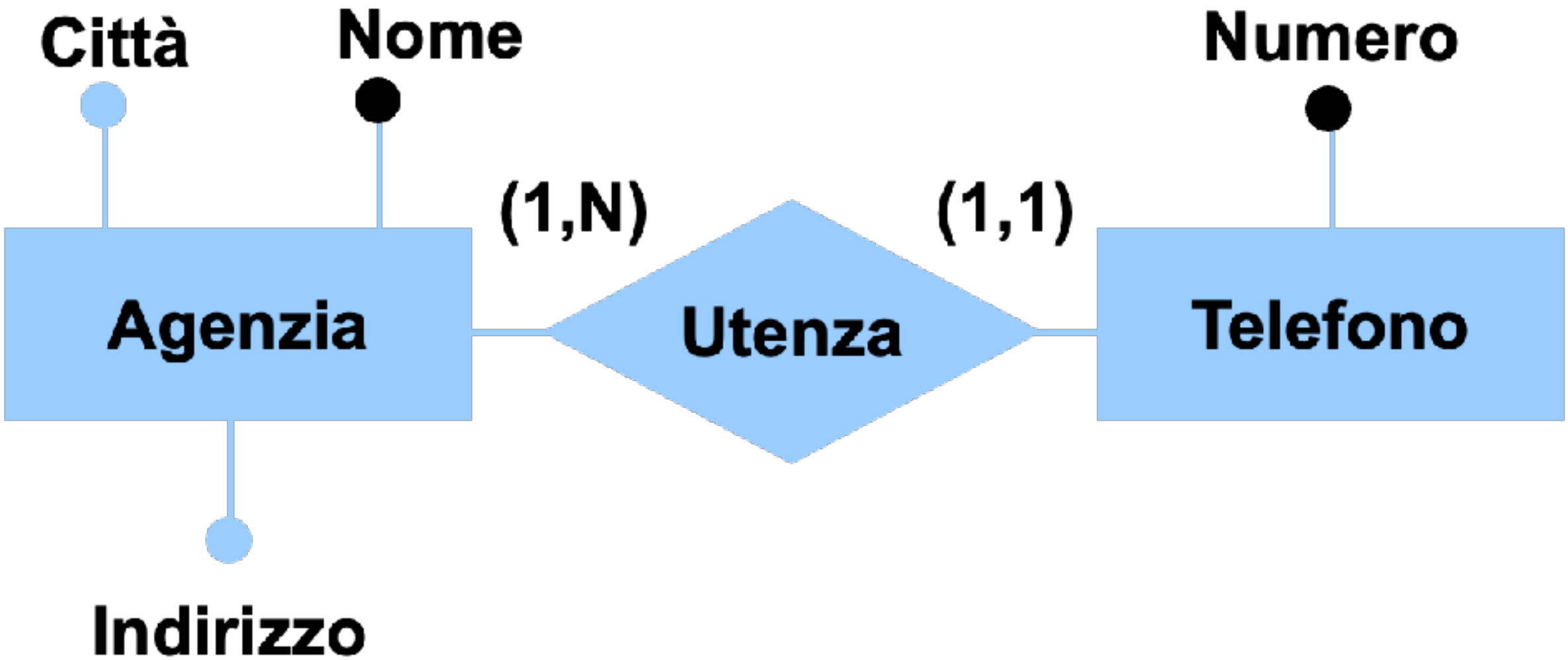
Partizione orizzontale di *relationship*



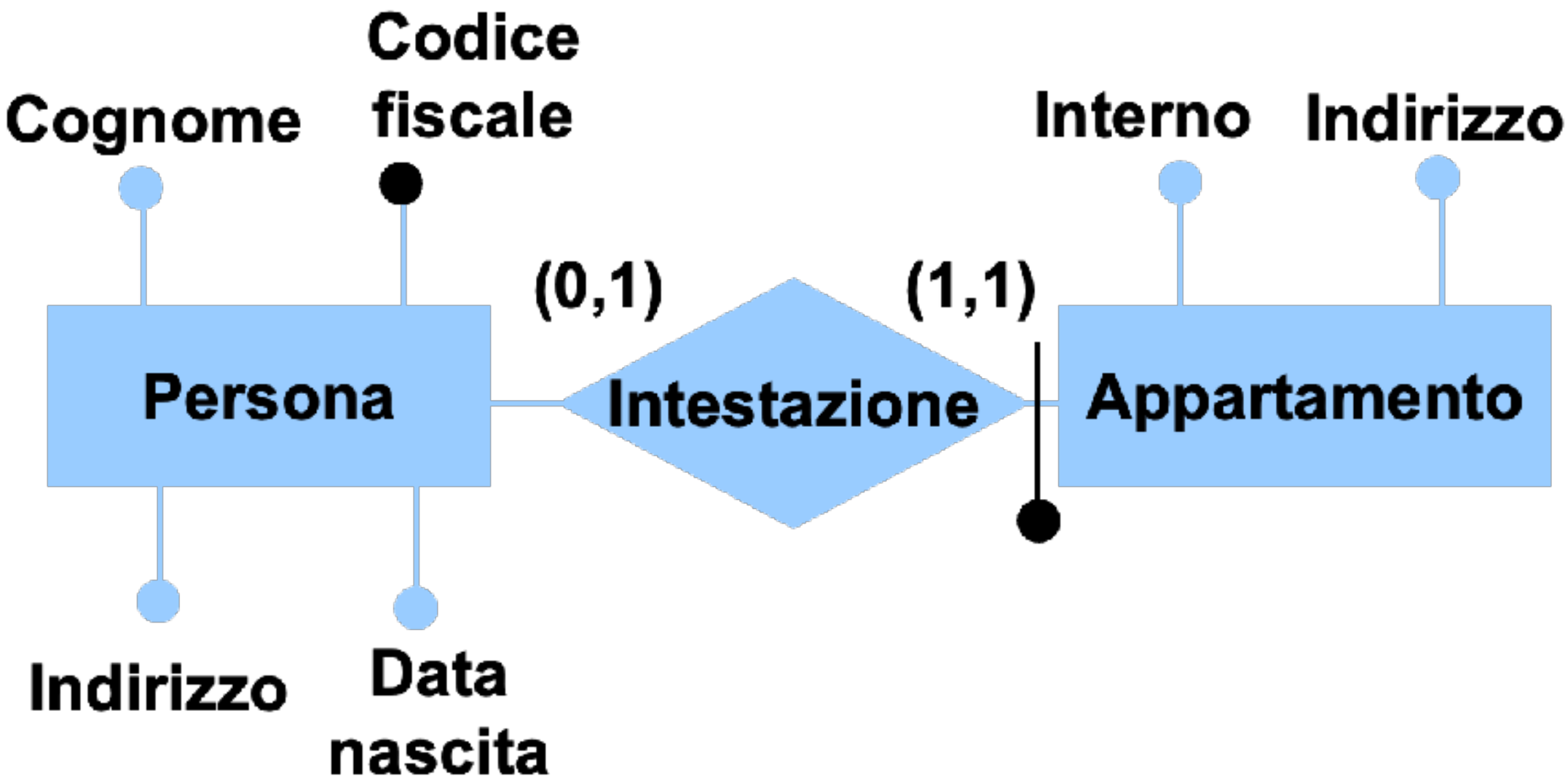
Eliminazione di attributi multivalore



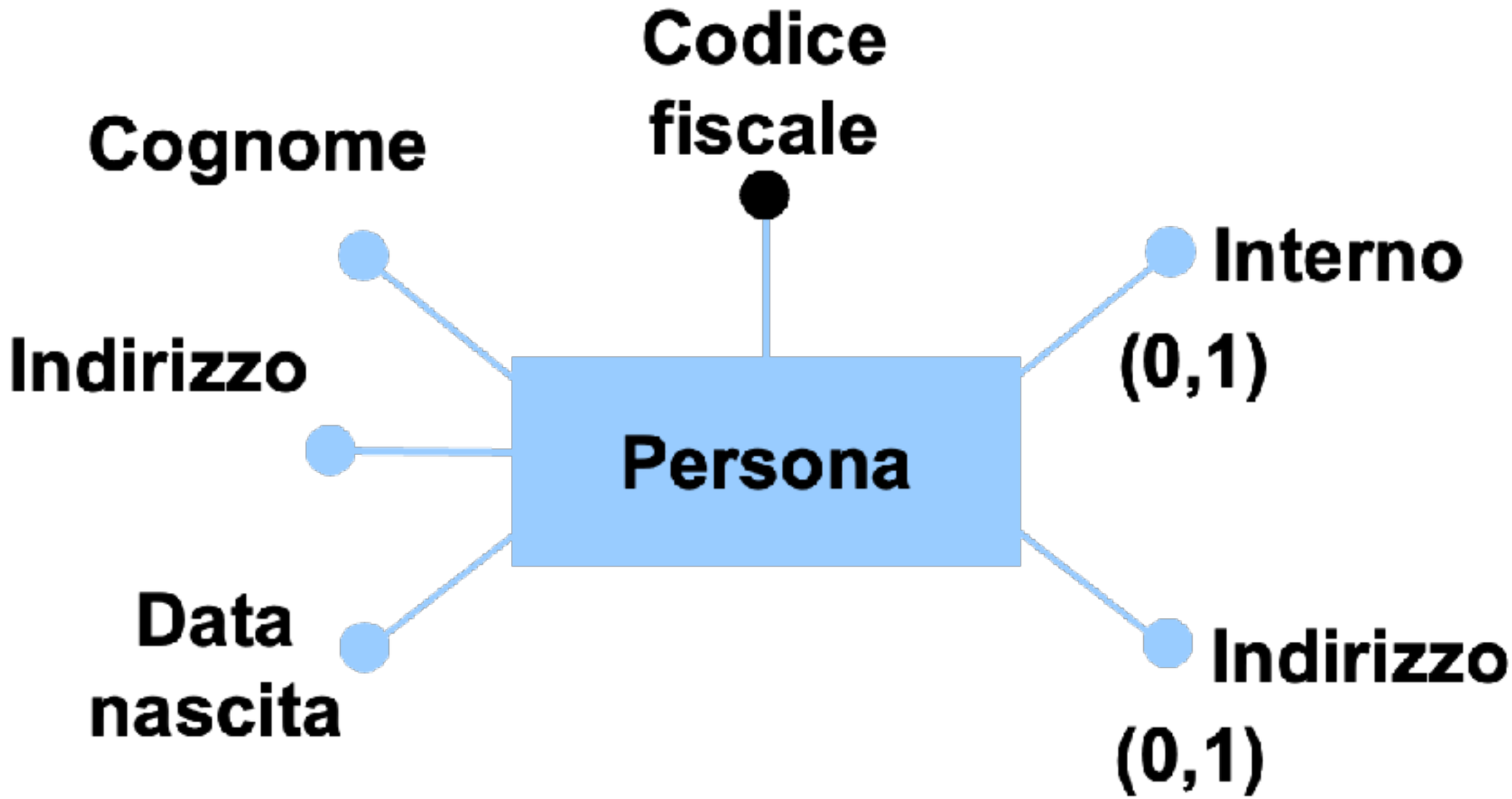
Eliminazione di attributi multivalore



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**

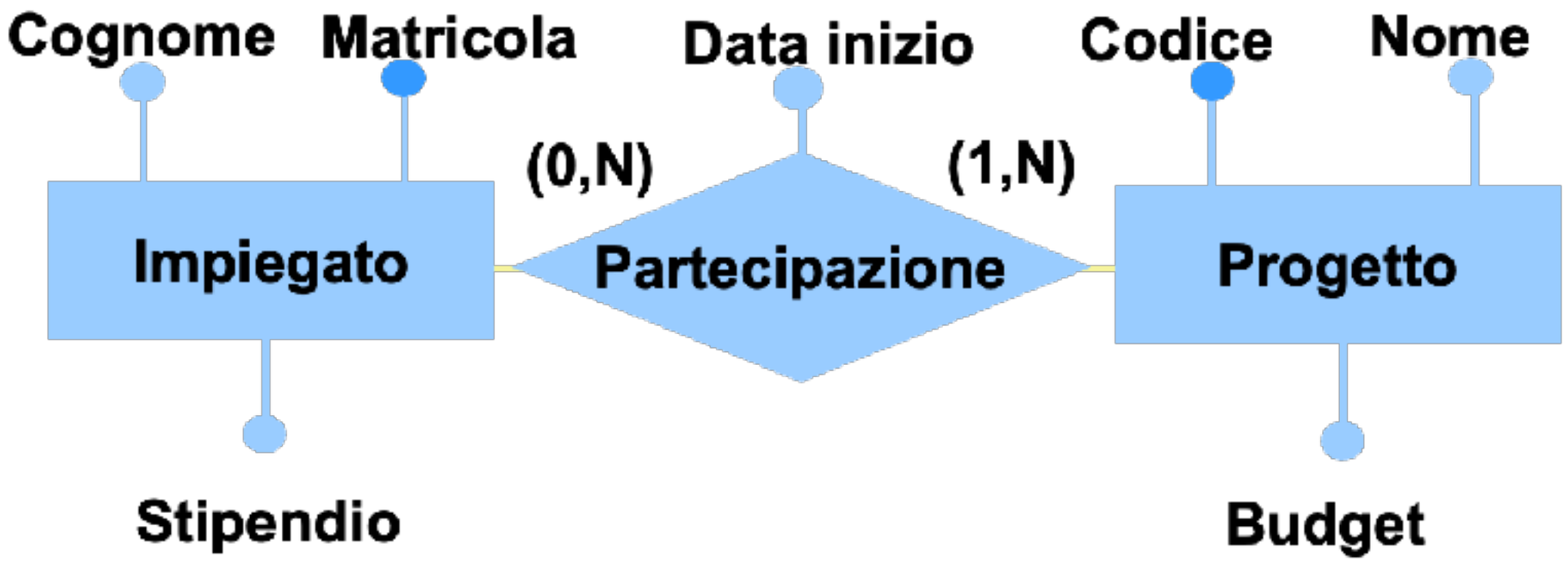
Scelta degli identificatori principali

- **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

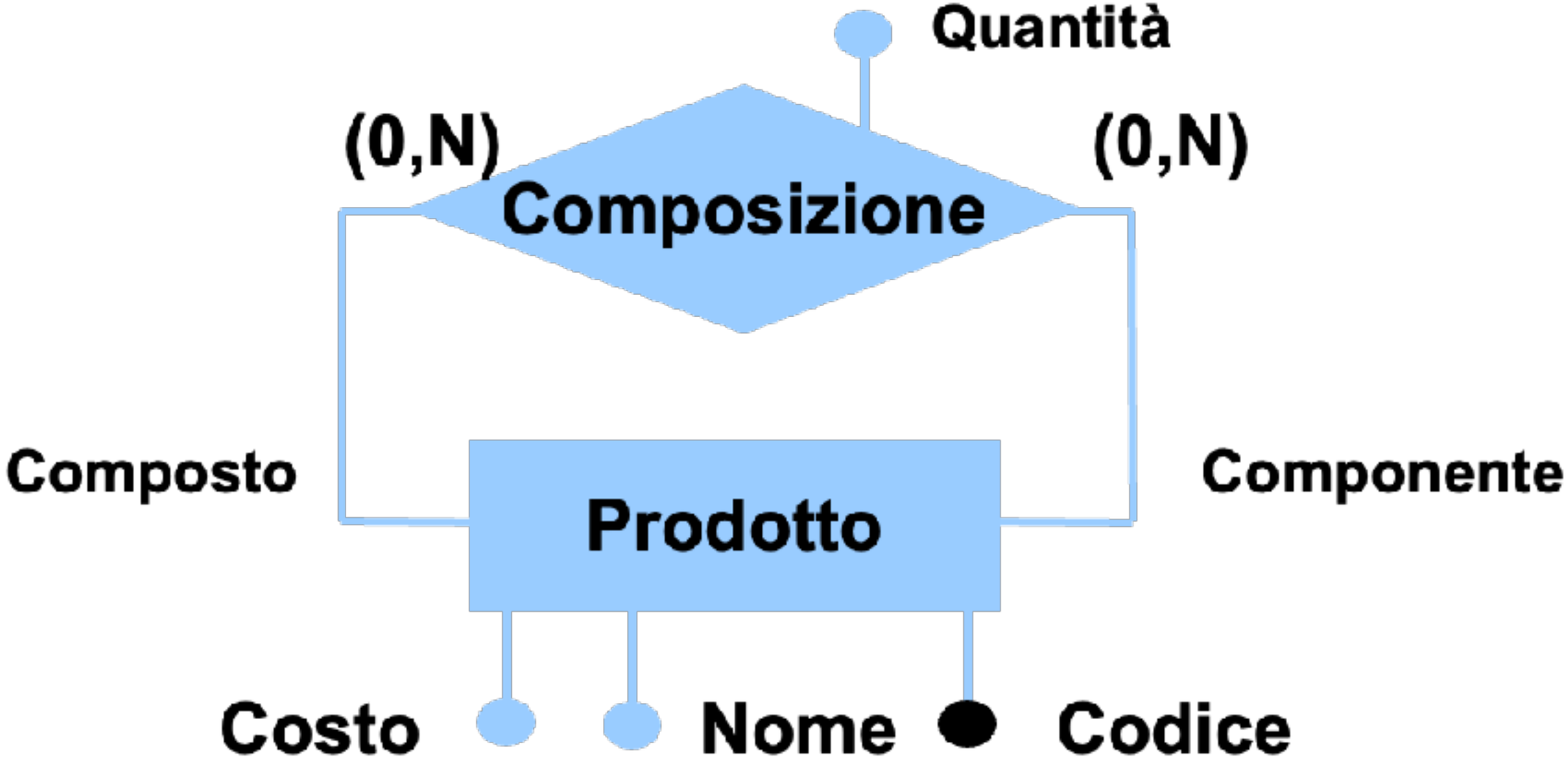


- **Impiegato**(<u>Matricola</u>, Cognome, Stipendio)
- **Progetto**(<u>Codice</u>, Nome, Budget)
- **Partecipazione**(<u>Matricola</u>, <u>Codice</u>, DataInizio)
- **Vincoli di integrità referenziale** fra:
  - <u>Matricola</u> in **Partecipazione** e (la chiave di) **Impiegato**
  - <u>Codice</u>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

Entità e relationship molti a molti

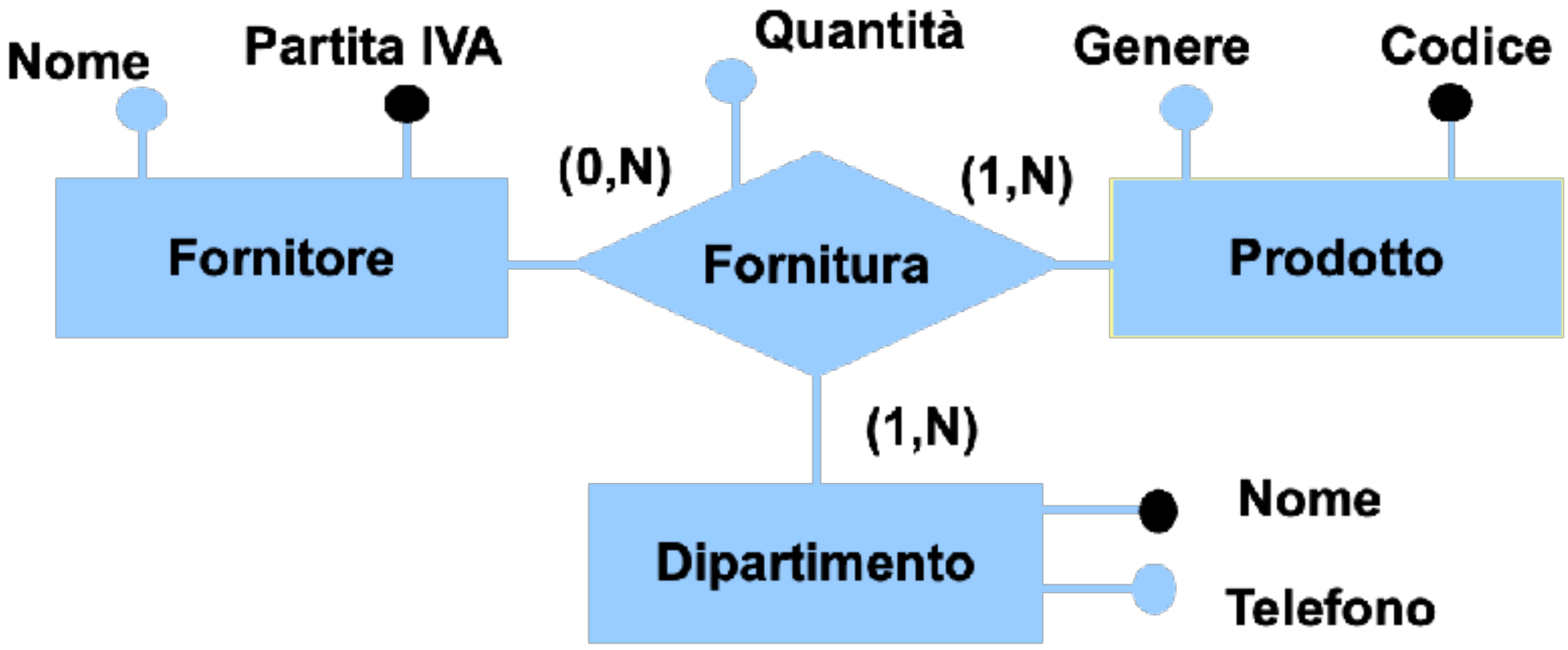
- Nomi più espressivi per gli attributi della chiave della relazione che rappresenta la relationship:
  - **Impiegato**(Matricola, Cognome, Stipendio)
  - **Progetto**(Codice, Nome, Budget)
  - **Partecipazione**(Matricola, Codice, DataInizio)
  - **Partecipazione**(Impiegato, Progetto, DataInizio)

Relationship Ricorsive



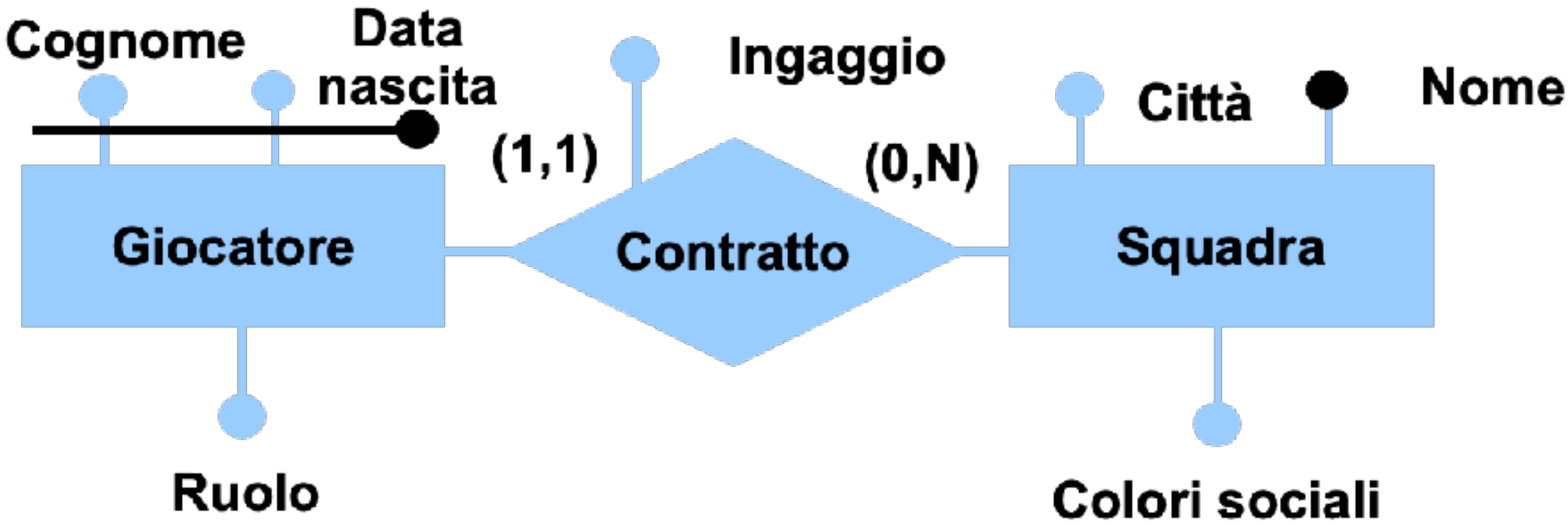
- **Prodotto**(<u>Codice</u>, Nome, Costo)
- **Composizione**(<u>Composto</u>, <u>Componente</u>, Quantità)

Relationship n-arie



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

Relationship uno a molti

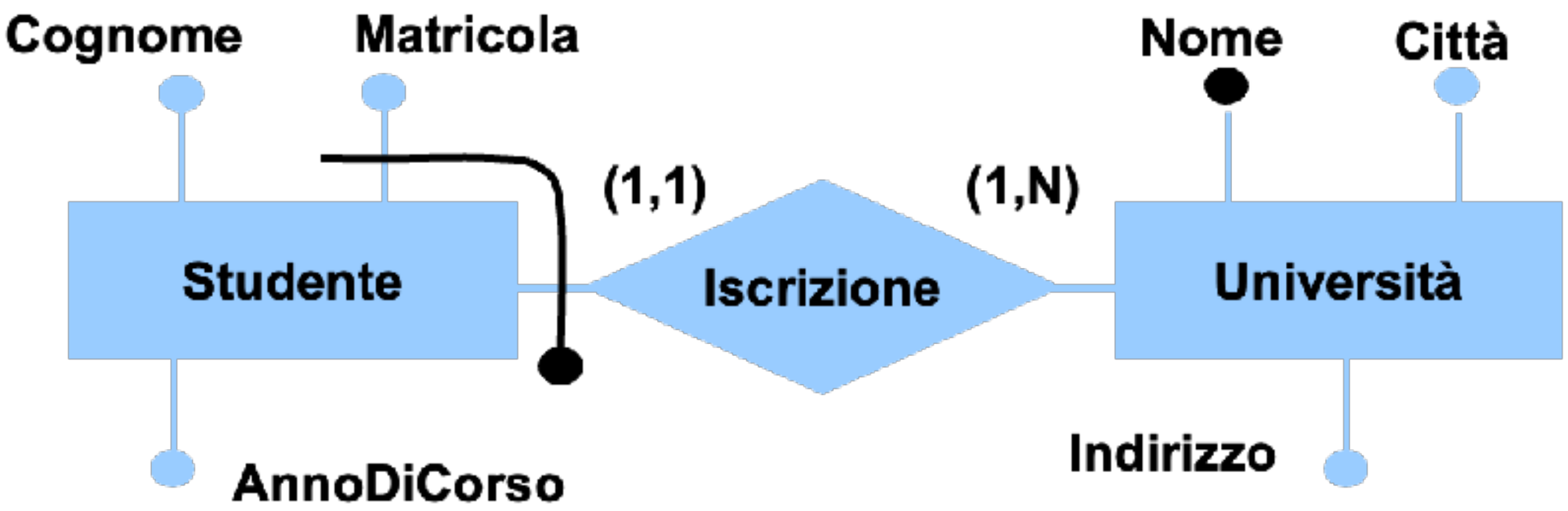


- **Giocatore**(<u>Cognome</u>, <u>DataNascita</u>, Ruolo)
- **Contratto**(<u>CognGiocatore</u>, <u>DataNascG</u>, <u>Squadra</u>, Ingaggio)
- **Squadra**(<u>Nome</u>, Città, ColoriSociali)
- È corretto?

Soluzione più compatta

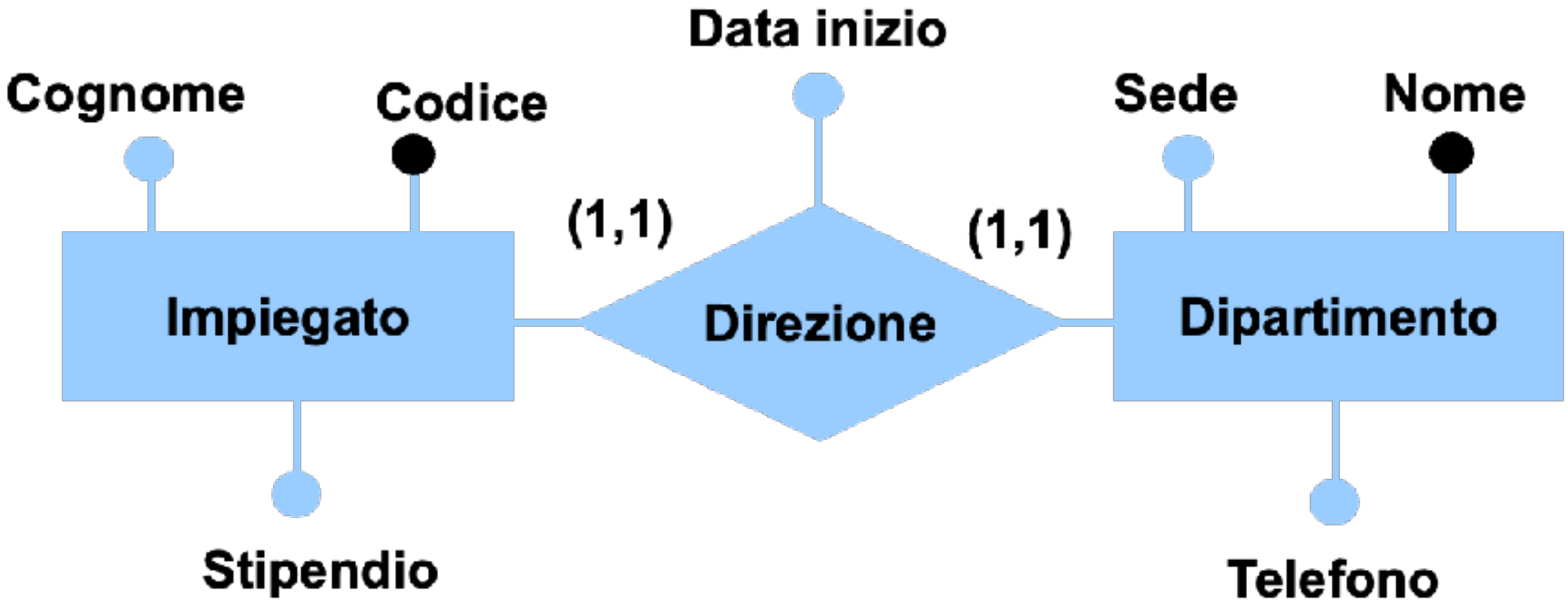
- **Giocatore**(Cognome, DataNasc, Ruolo, Squadra, Ingaggio)
- **Squadra**(Nome, Città, ColoriSociali)
- Con vincolo di integrità referenziale fra **Squadra** in **Giocatore** e (la chiave di) **Squadra**
- Se la cardinalità minima della *relationship* è 0, allora **Squadra** in **Giocatore** deve ammettere 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



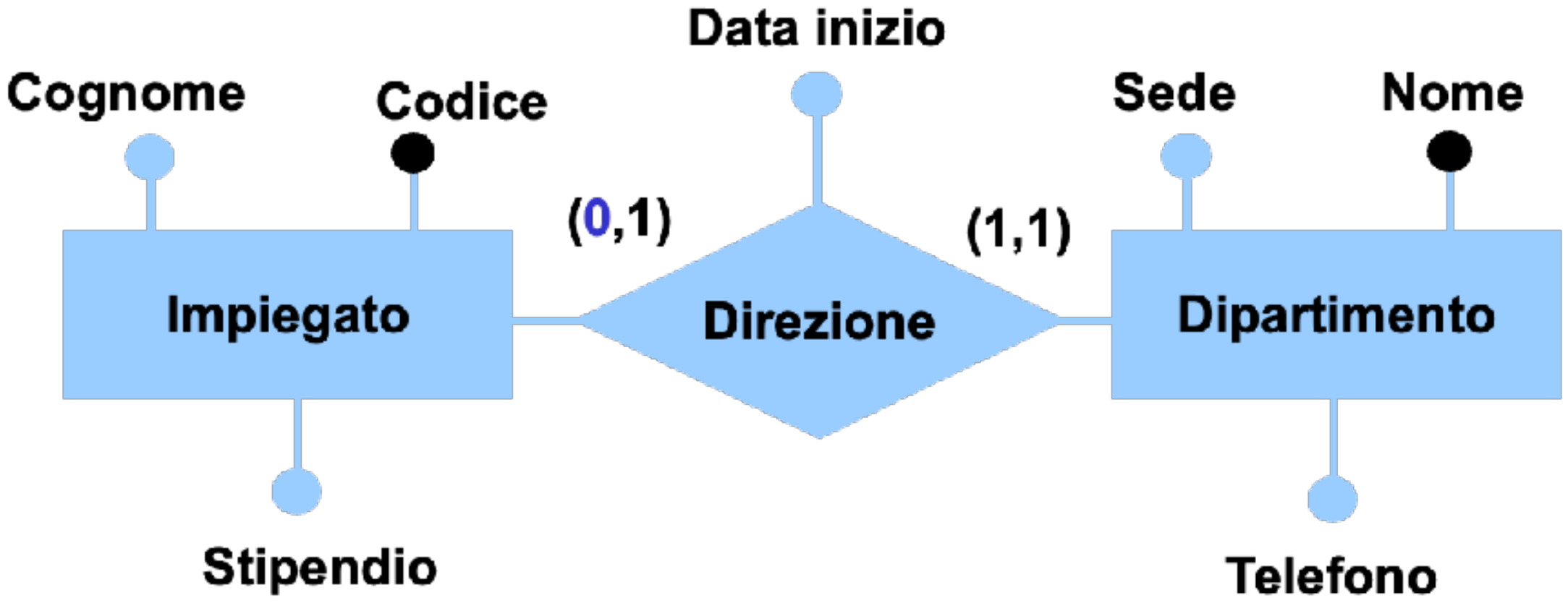
- **Studente**(<u>Matricola</u>, <u>Università</u>, Cognome, AnnoDiCorso)
- **Università**(<u>Nome</u>, Città, Indirizzo)
- con vincolo ...

Relationship uno a uno



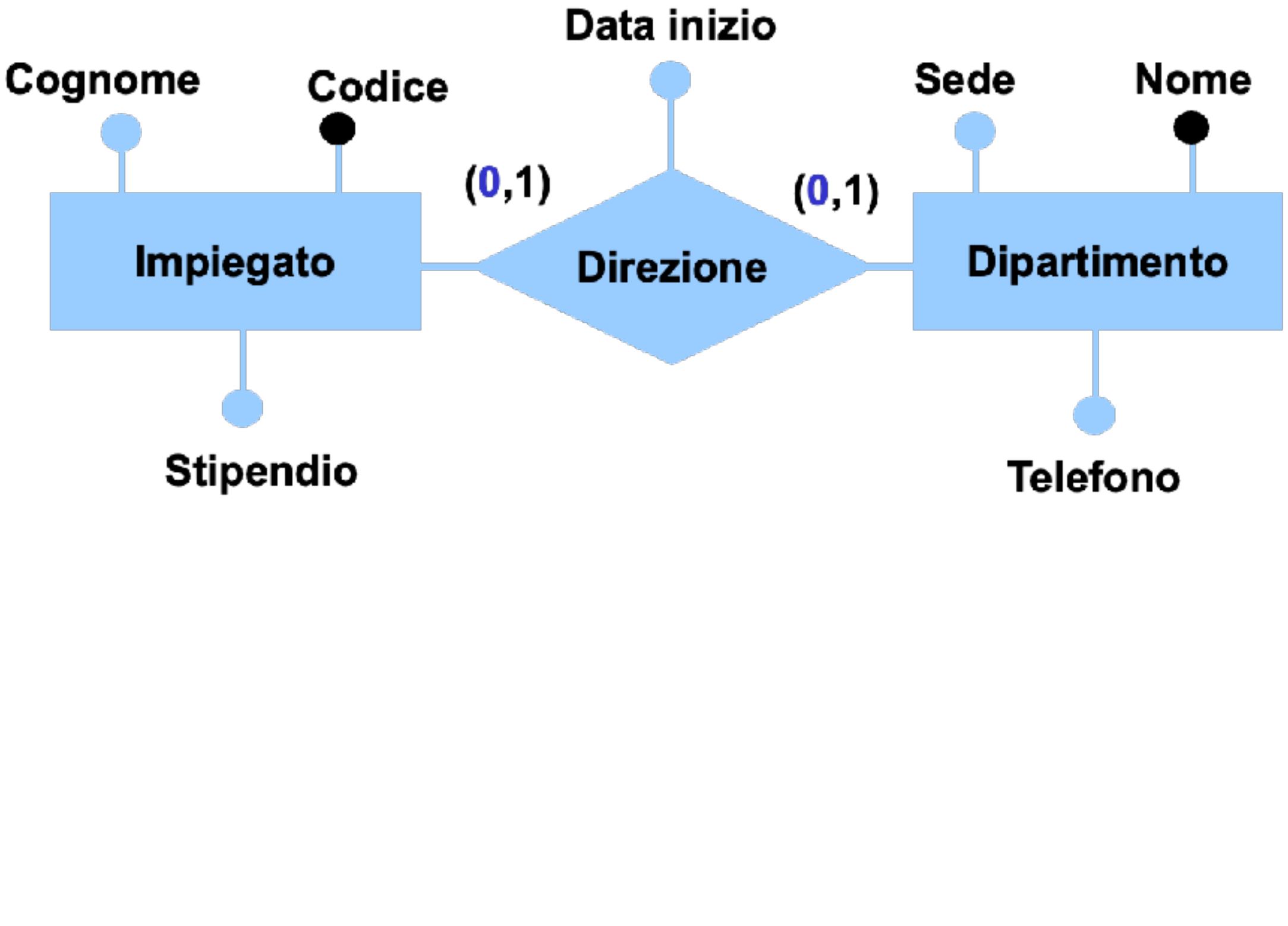
- Varie possibilità:
  - fondere da una parte o dall'altra
  - fondere tutto?

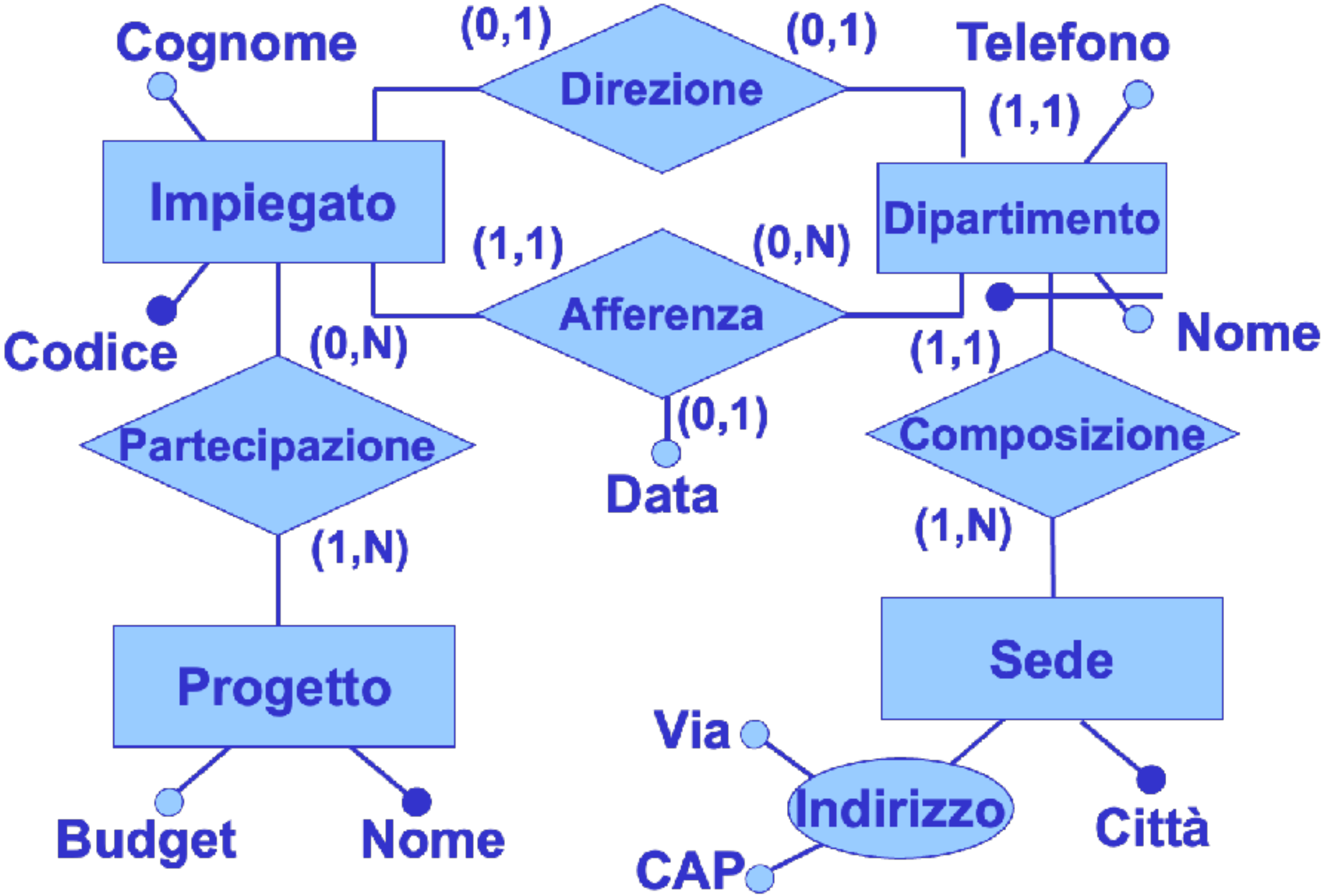
Un caso privilegiato



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

Un altro caso

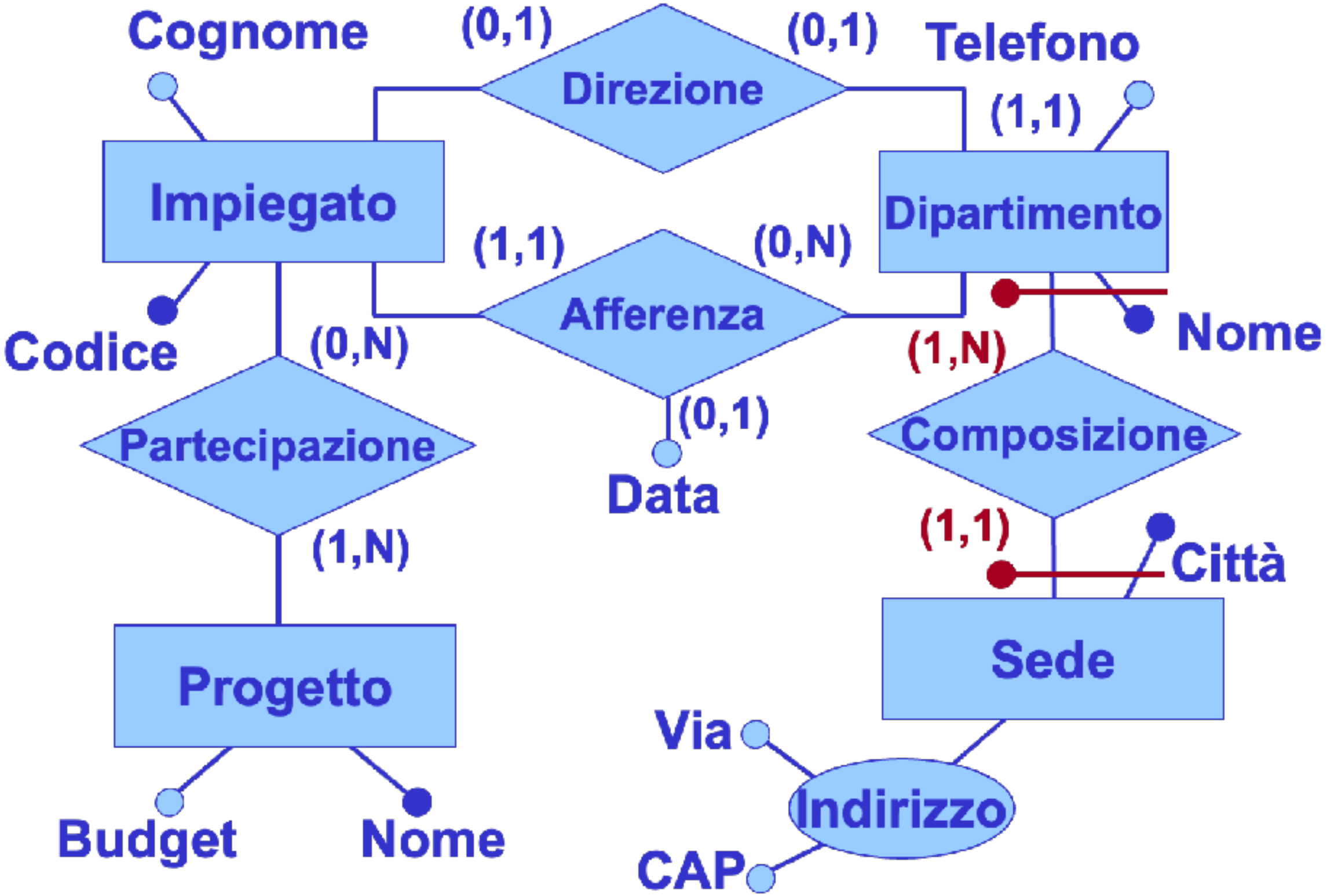



Cognome
(0,1)
Direzione
(0,1)
Telefono
(1,1)
Impiegato
Dipartimento
(1,1)
Afferenza
(0,N)
Codice
(0,N)
(1,1)
Nome
Partecipazione
(0,1)
Composizione
Data
(1,N)
(1,N)
Progetto
Sede
Via
Indirizzo
Città
CAP
Budget
Nome

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 ...


Cognome
(0,1)
Direzione
(0,1)
Telefono
(1,1)
Impiegato
Dipartimento
(1,1)
(0,N)
Afferenza
Codice
(0,N)
(1,N)
Nome
Partecipazione
(0,1)
Composizione
Data
(1,1)
Città
(1,N)
Sede
Progetto
Via
Indirizzo
Budget
Nome
CAP