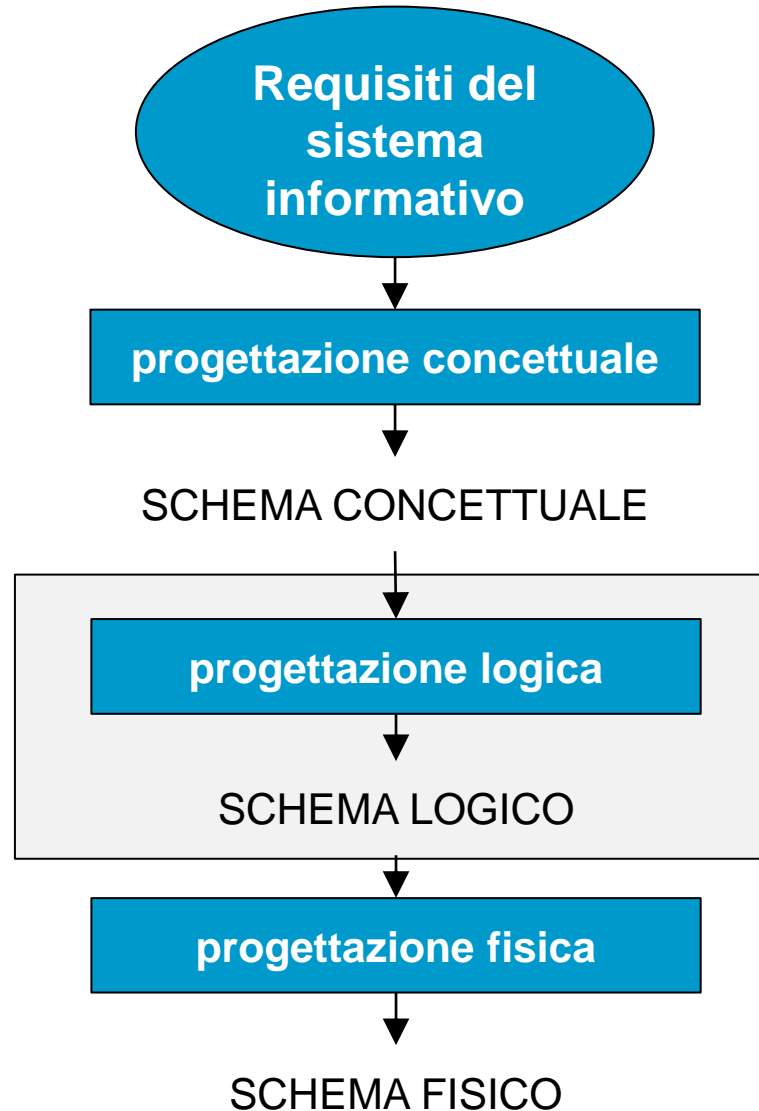




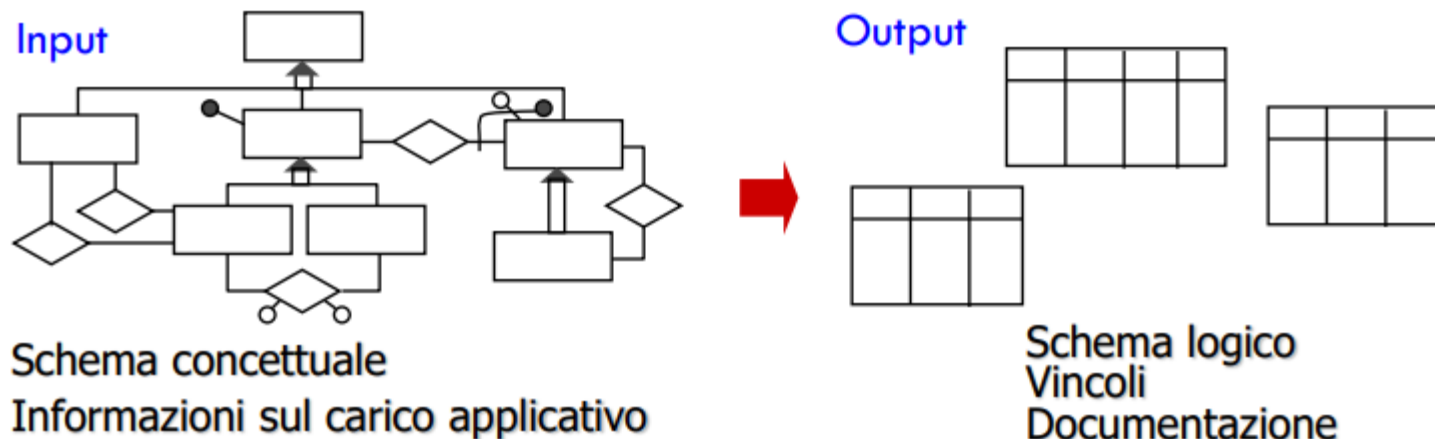
# **Progettazione logica**

# Il secondo passo..



# Obiettivi

- ❑ Obiettivo della fase di progettazione logica è pervenire, a partire dallo schema concettuale, a uno schema logico che rappresenti **in modo fedele** i concetti e i requisiti analizzati e che sia, al tempo stesso, **“efficiente”**.



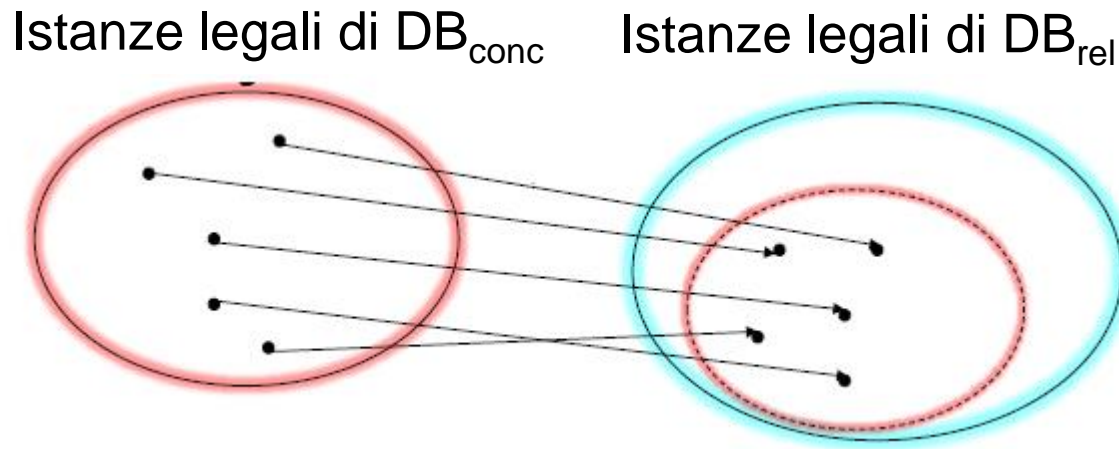
- ❑ L'**efficienza** è legata alle **prestazioni**, ma poiché queste non sono valutabili precisamente né a livello concettuale né a livello logico, si ricorre all'impiego di **indicatori semplificati**.
- ❑ L'input per questa fase è lo schema E/R e il carico di lavoro stimato del database, in termini di quantità di dati e requisiti operativi.

# Progettazione logica “fedele” = equivalenza

- ❑ Che cosa s'intende precisamente quando si dice che uno schema relazionale  $DB_{rel}$  rappresenta “fedelmente” uno schema concettuale (E/R)  $DB_{conc}$ ?
- ❑ Intuitivamente “fedeltà” vuol dire che mediante  $DB_{rel}$  possiamo rappresentare esattamente le medesime informazioni documentate con lo schema  $DB_{conc}$  (*possiamo memorizzare gli stessi dati*).
- ❑ Più precisamente “fedeltà” significa che i due schemi sono equivalenti dal punto di vista della loro capacità informativa.
- ❑ Il concetto di capacità informativa ha diverse definizioni, ma per i nostri scopi può essere considerato equivalente all'insieme delle istanze legali di uno schema, indicato con  $Sat(DB)$  e dunque:
- ❑  $DB_{rel}$  e  $DB_{conc}$  sono equivalenti se  $Sat(DB_{conc}) = Sat(DB_{rel})$ .

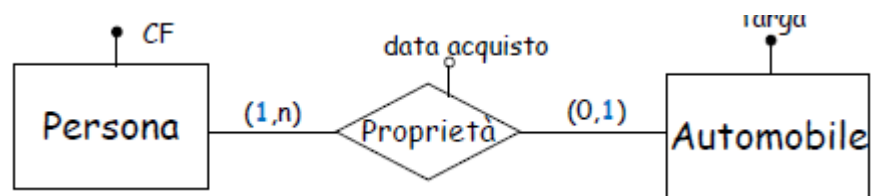
# Progettazione che preserva l'informazione

- La definizione intuitivamente asserisce che lo schema relazionale può contenere i dati dello schema E/R (**totalità**) e che si può “ritornare indietro” (**iniettività**).



# Perché ciò non basta

- Si consideri il seguente schema E/R:



- e lo schema relazionale:

**Persona** (CF)

**Auto** (Targa)

**Proprietà** (CF, Targa, DataAcquisto)

FK: CF REFERENCES Persone

FK: Targa REFERENCES Auto

- La traduzione preserva l'informazione, ma esistono infinite istanze che sono legali rispetto a  $DB_{rel}$  e che non lo sono per  $DB_{conc}$

**Persona**

CF
BLGSTR71B22
FDLNNR66M45
<b>BSZNTN82L27</b>

**Proprietà**

CF	Targa	DataAcquisto
BLGSTR71B22	<b>CT 001 MJ</b>	12/08/2004
FDLNNR66M45	<b>CT 001 MJ</b>	15/07/2003

# Come agire in pratica?

- ❑ La definizione data di equivalenza non è “operativa”, in quanto non dice nulla su come debba essere fatta una traduzione che garantisca l'equivalenza degli schemi.
- ❑ Tuttavia può essere usata “localmente”:  
in pratica la traduzione da schema E/R a schema relazionale avviene operando una **sequenza di trasformazioni/traduzioni semplici**, per ognuna delle quali è altrettanto semplice rispettare regole che garantiscono l'equivalenza.
- ❑ Per quanto visto, possiamo dividere queste regole in:
  - **regole che preservano l'informazione (regole sulla “struttura”);**
  - **regole aggiuntive che garantiscono l'equivalenza (regole sui vincoli).**
- ❑ L'equivalenza può comunque essere solo in parte garantita dal DDL di SQL, infatti alcuni vincoli non possono essere direttamente espressi in SQL.

# Fasi della progettazione logica

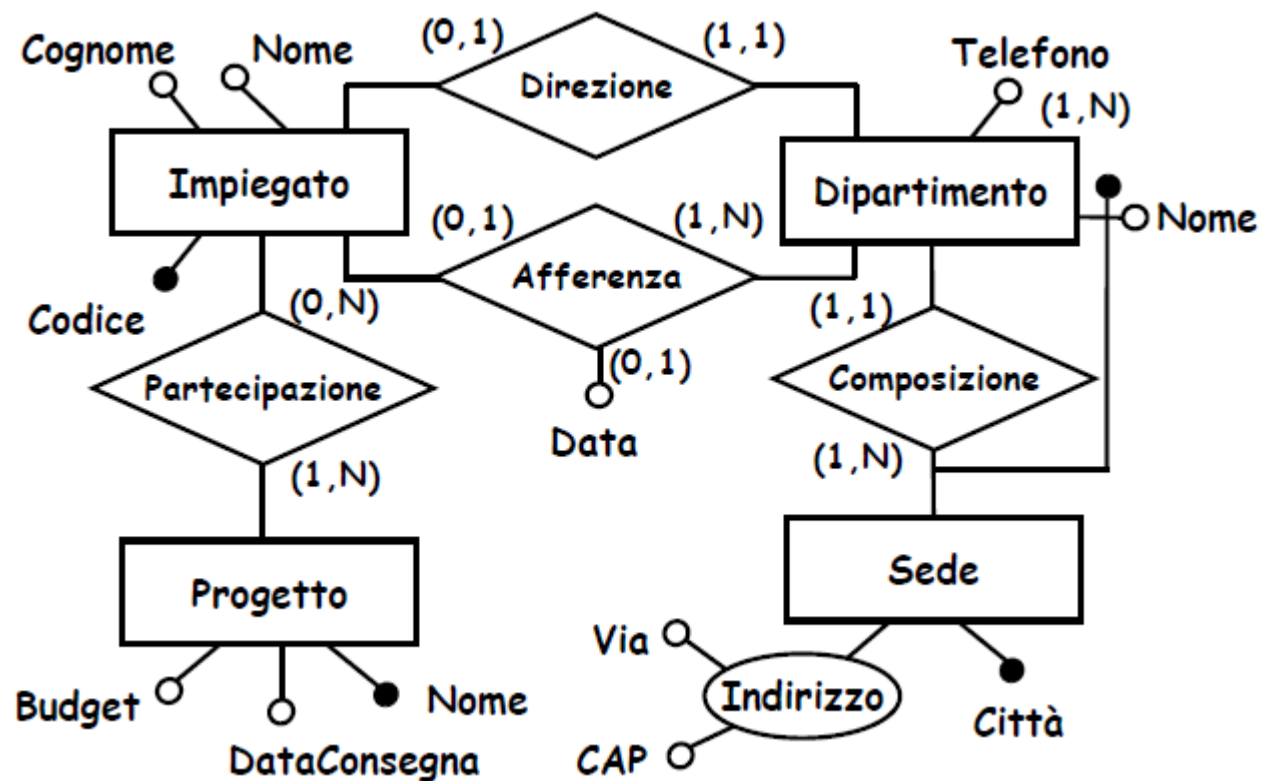
- La progettazione logica può essere articolata in due fasi principali:
  - **Ristrutturazione:** eliminazione dallo schema E/R dei costrutti che non possono essere direttamente rappresentati nel modello logico target (**relazionale nel nostro caso**):
    - eliminazione degli attributi multivalore;
    - eliminazione delle gerarchie di generalizzazione;
    - partizionamento/accorpamento di entità e associazioni;
    - scelta degli identificatori principali.
  - **Traduzione:** si mappano i costrutti residui in elementi del modello relazionale.



# Fase di ristrutturazione

- ❑ Si pone l'obiettivo di **semplificare la traduzione** e “**ottimizzare**” le prestazioni.
- ❑ Per confrontare tra loro diverse alternative bisogna conoscere, almeno in maniera approssimativa, il “**carico di lavoro**”, ovvero:
  - le principali **operazioni** che la base dati dovrà supportare;
  - i **volumi dei dati** in gioco.
- ❑ **Regola 80-20: il 20% delle operazioni produce l'80% del carico.**
- ❑ Gli **indicatori** che deriviamo considerano due aspetti
  - **spazio**: numero di istanze (di entità e associazioni) previste;
  - **tempo**: numero di istanze visitate durante un'operazione.

# Schema di riferimento



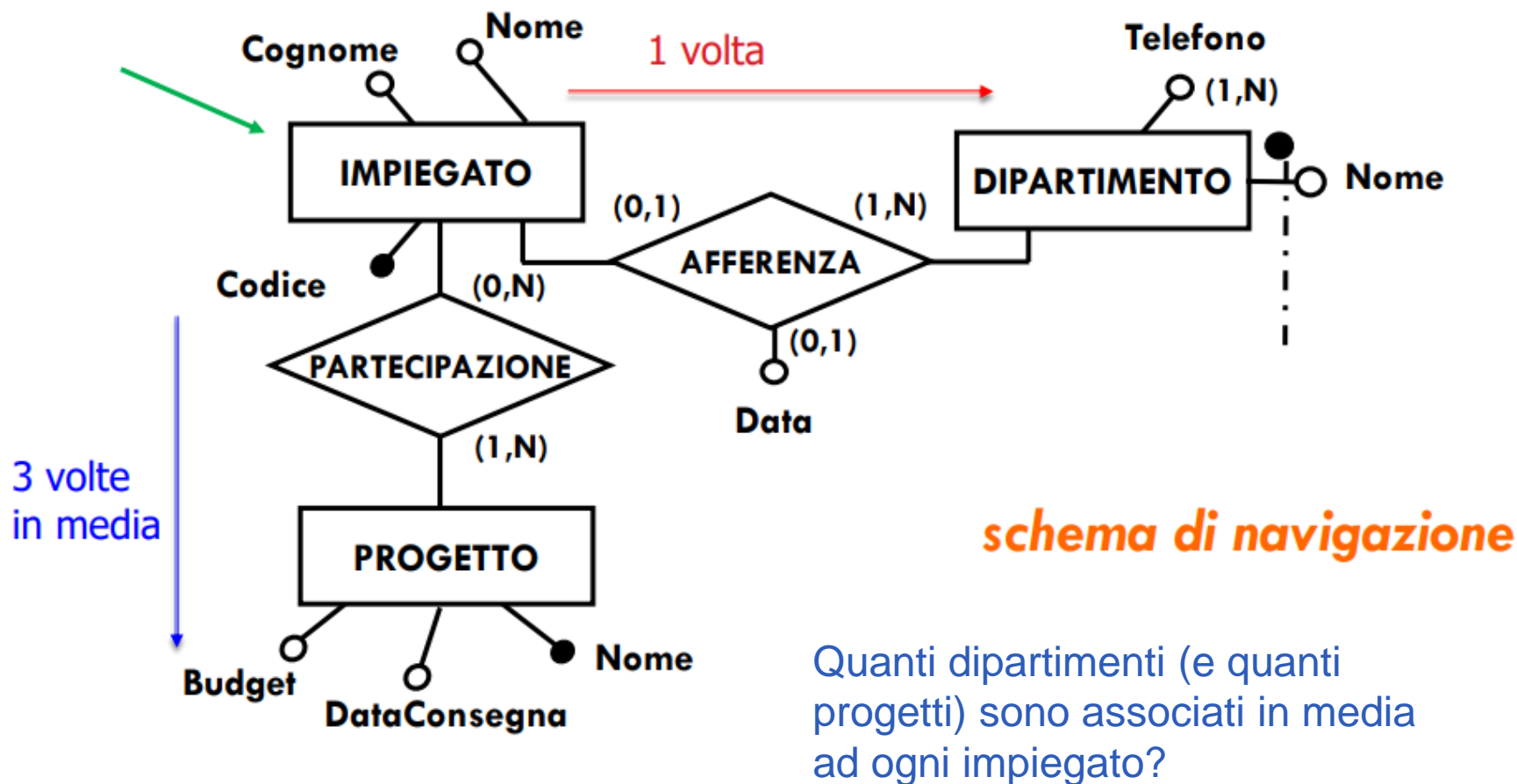
# Operazioni

- ❑ **Operazione 1:** assegna un impiegato a un progetto
- ❑ **Operazione 2:** trova i dati di un impiegato, del dipartimento nel quale lavora e dei progetti ai quali partecipa
- ❑ **Operazione 3:** trova i dati di tutti gli impiegati di un certo dipartimento
- ❑ **Operazione 4:** per ogni sede, trova i suoi dipartimenti con il cognome del direttore e l'elenco degli impiegati

Operazione	Tipo	Frequenza
Op. 1	I	50/giorno
Op. 2	I	100/giorno
Op. 3	I	10/giorno
Op. 4	B	2/settimana

# Esempio valutazione di costo

**Operazione 2:** Visualizzare tutti i dati di un *impiegato*, del *dipartimento* nel quale lavora e dei *progetti* ai quali partecipa.



# Volumi dei dati

- ❑ Numero di istanze per ogni entità e associazione dello schema
- ❑ Dimensione di ogni attributo
- ❑ Nella **tavola dei volumi** si riportano per tutti i concetti (entità e associazioni) i volumi previsti a regime

Concetto	Tipo	Volume
<Nome>	E/R	<dimensione>

- ❑ Per le associazioni il volume dipende da
  - numero di istanze coinvolte nella associazione
  - il numero (medio) di partecipazioni di una istanza di entità alle istanze di associazione (**dipende dalla cardinalità delle associazioni**)

# Tavola dei volumi

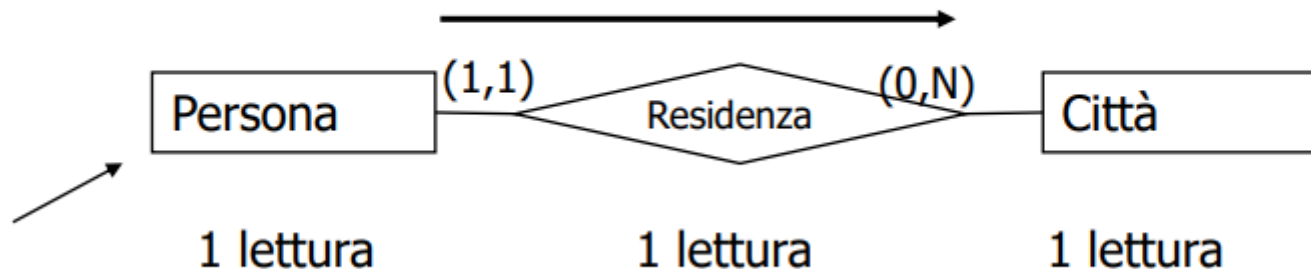
Concetto	Costrutto	Volume
SEDE	E	10
DIPARTIMENTO	E	80
IMPIEGATO	E	2000
PROGETTO	E	500
COMPOSIZIONE	A	80
AFFERENZA	A	1900
DIREZIONE	A	80
PARTECIPAZIONE	A	6000

- **Composizione:** è pari al numero di Dipartimenti (1 Dipartimento - 1 Sede)
- **Afferenza:** è paragonabile (leggermente inferiore) al numero di Impiegati
- **Partecipazione:** si assume che in media un Impiegato partecipi a 3 Progetti
- **Direzione:** ogni Dipartimento ha un Direttore

# Schema di navigazione

- ❑ Lo **schema di navigazione** descrive i dati coinvolti in un'operazione
- ❑ Corrisponde al frammento dello schema ER interessato all'operazione sul quale viene disegnato il cammino logico per accedere alle informazioni di interesse

**Operazione:** data una Persona trovare la città di Residenza



# Tavola degli accessi

- ❑ Con lo schema di navigazione si può fare una stima del costo di un'operazione contando il numero di accessi alle istanze di entità e associazioni
- ❑ Il risultato può essere riassunto in una tavola degli accessi

Concetto	Tipo	Accessi	Tipo
<Name>	E/A	<number>	S/L

- ❑ Il tipo distingue gli accessi in scrittura (S) e in lettura (L)
- ❑ Le operazioni di scrittura sono in genere più onerose (esecuzione in modo esclusivo, aggiornamento degli indici)
- ❑ Il costo di una scrittura viene considerato pari a 2 operazioni (O), una lettura pari a 1



# Esempio tavola degli accessi

- Il numero delle istanze si ricava dalla tavola dei volumi mediante semplici operazioni (assumendo uniformità nella distribuzione dei valori):
  - ad esempio in media ogni impiegato partecipa a  $6000/2000 = 3$  progetti.



## Operazione 2:

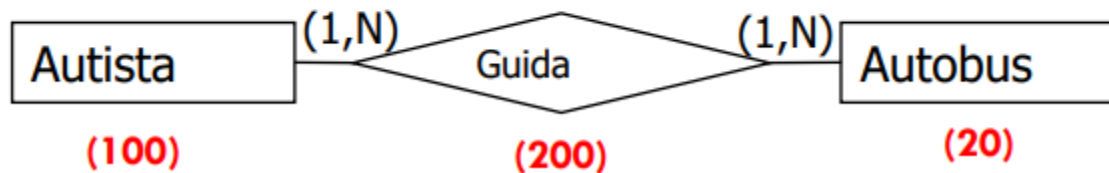
- Impiegato: 1 accesso
  - Afferenza: 1 accesso (ogni Impiegato afferisce al più a un Dipartimento)
  - Dipartimento: 1 accesso
  - Partecipazione: 3 accessi
  - Progetto: 3 accessi
- Tutti gli accessi sono in lettura

Concetto	Costrutto	Accessi	Tipo
IMPIEGATO	E	1	L
AFFERENZA	A	1	L
DIPARTIMENTO	E	1	L
PARTECIPAZIONE	A	3	L
PROGETTO	E	3	L

# Esercizio 1

- Stimare il numero degli accessi

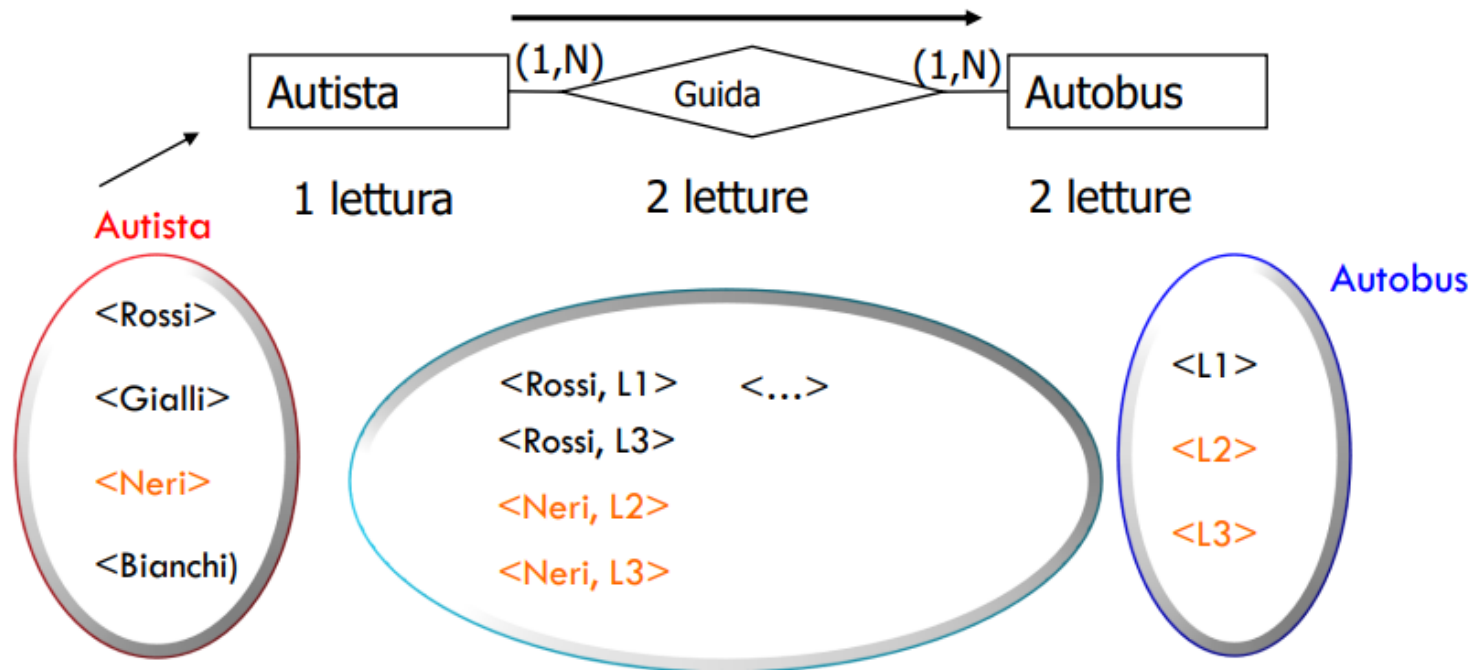
**operazione:** dato un autista trovare gli autobus che guida



# Esercizio 1 - soluzione

- Stimare il numero degli accessi

**operazione:** dato un autista trovare gli autobus che guida

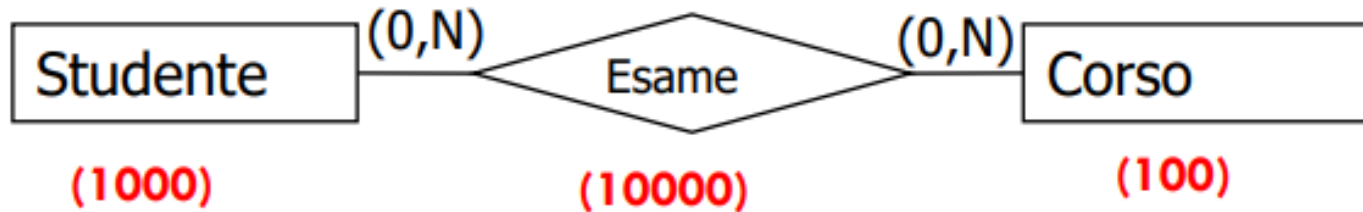


- Per calcolare il numero di accessi a Guida e Autobus, è necessario conoscere il numero medio di istanze in Guida per Autista ( $200/100=2$ )

## Esercizio 2

- Stimare il numero degli accessi

**Operazione:** Stampare il curriculum di uno Studente



# Esercizio 2 - soluzione

- Stimare il numero degli accessi

**Operazione:** Stampare il curriculum di uno Studente



Il numero medio degli esami sostenuti dagli studenti è  $10000/1000 = 10$

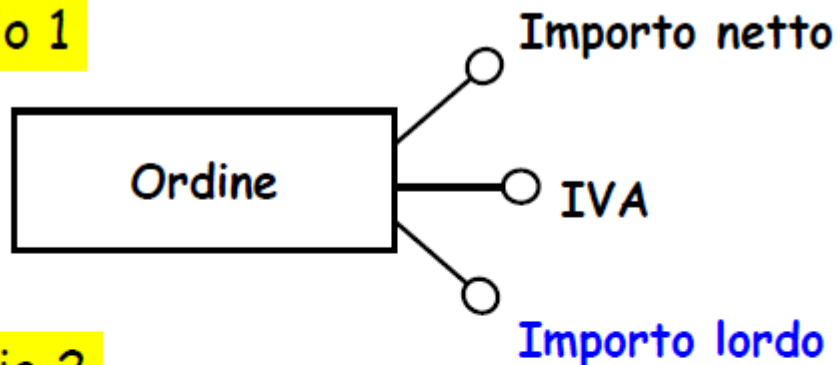
Concetto	Tipo	Accesso	Tipo
Studente	E	1	L
Esame	A	10	L
Corso	E	10	L
<b>TOTALE</b>		<b>21</b>	O

# Analisi delle ridondanze

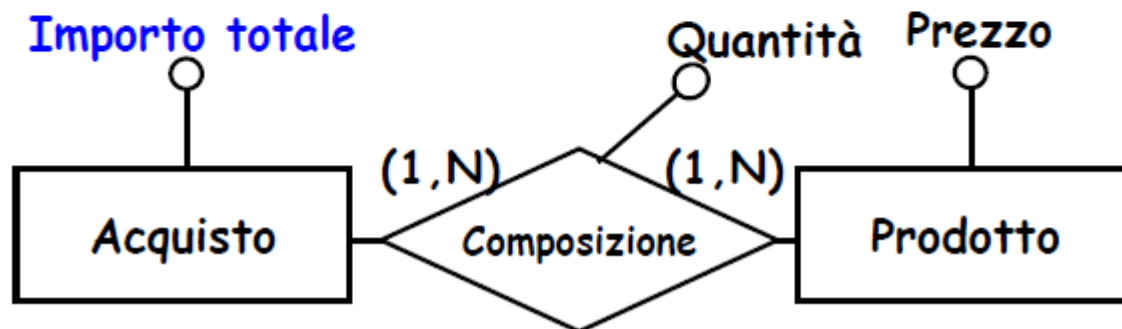
- ❑ Una **ridondanza** in uno schema E-R è **un'informazione significativa** ma **derivabile** da altre.
- ❑ In questa fase si decide se eliminare o meno le ridondanze eventualmente presenti; **è quindi comunque importante averle individuate in fase di progettazione concettuale!**
- ❑ Se si mantiene una ridondanza
  - **si semplificano** alcune interrogazioni, ma
  - **si appesantiscono** gli aggiornamenti e
  - **si occupa maggior spazio.**
- ❑ Le possibili ridondanze riguardano
  - **attributi derivabili** da altri attributi;
  - associazioni derivabili dalla composizione di altre associazioni (**presenza di cicli**).

# Attributi derivabili

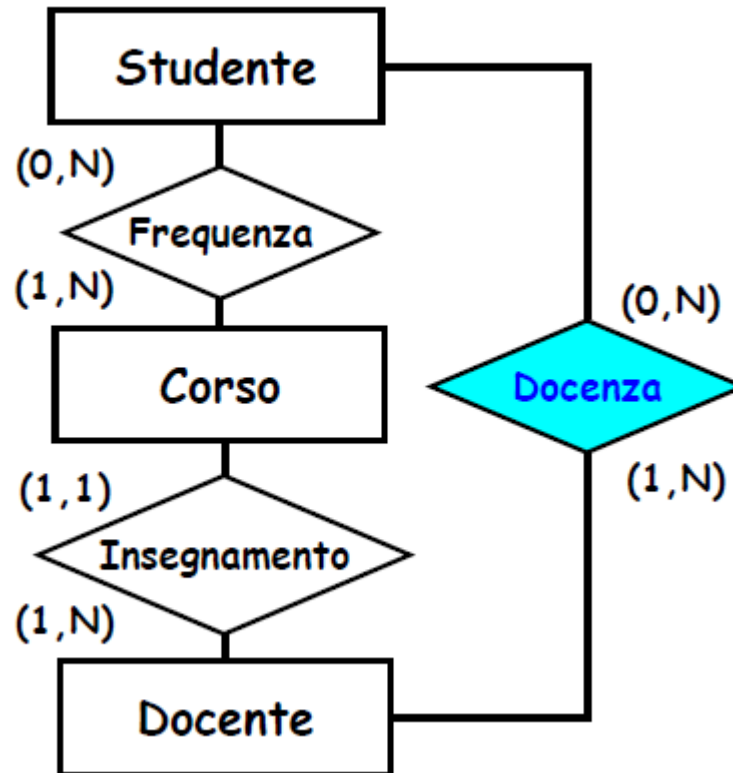
Esempio 1



Esempio 2



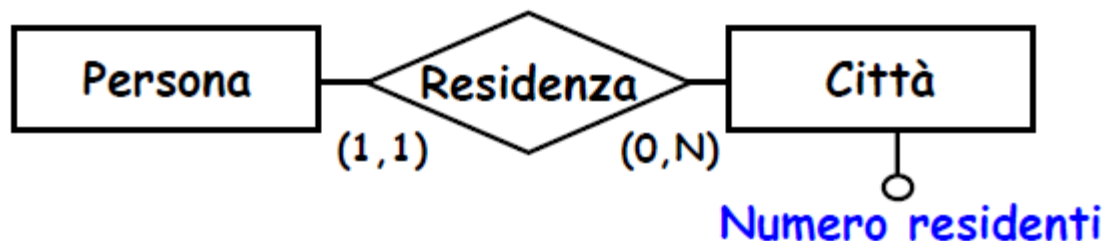
# Associazioni ridondanti





# Esempio d'analisi di una ridondanza

- L'attributo Numero residenti è derivabile da una operazione di conteggio delle istanze di persona residenti in una città



*tabella dei volumi*

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

# Le operazioni...

- ❑ Si considerano innanzitutto **le operazioni influenzate dalla ridondanza**, considerando anche le loro **frequenze di esecuzione**:
- ❑ **operazione 1**: inserisci una nuova persona con la relativa città di residenza (**500 volte al giorno**);
- ❑ **operazione 2**: visualizza tutti i dati di una città (incluso il numero di residenti) (**2 volte al giorno**);
- ❑ e si costruiscono le **tavole degli accessi**.

# ...in presenza di ridondanza...

## Operazione 1

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

## Operazione 2

Concetto	Costrutto	Accessi	Tipo
Città	Entità	1	L

# ...in assenza di ridondanza

## Operazione 1

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

## Operazione 2

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

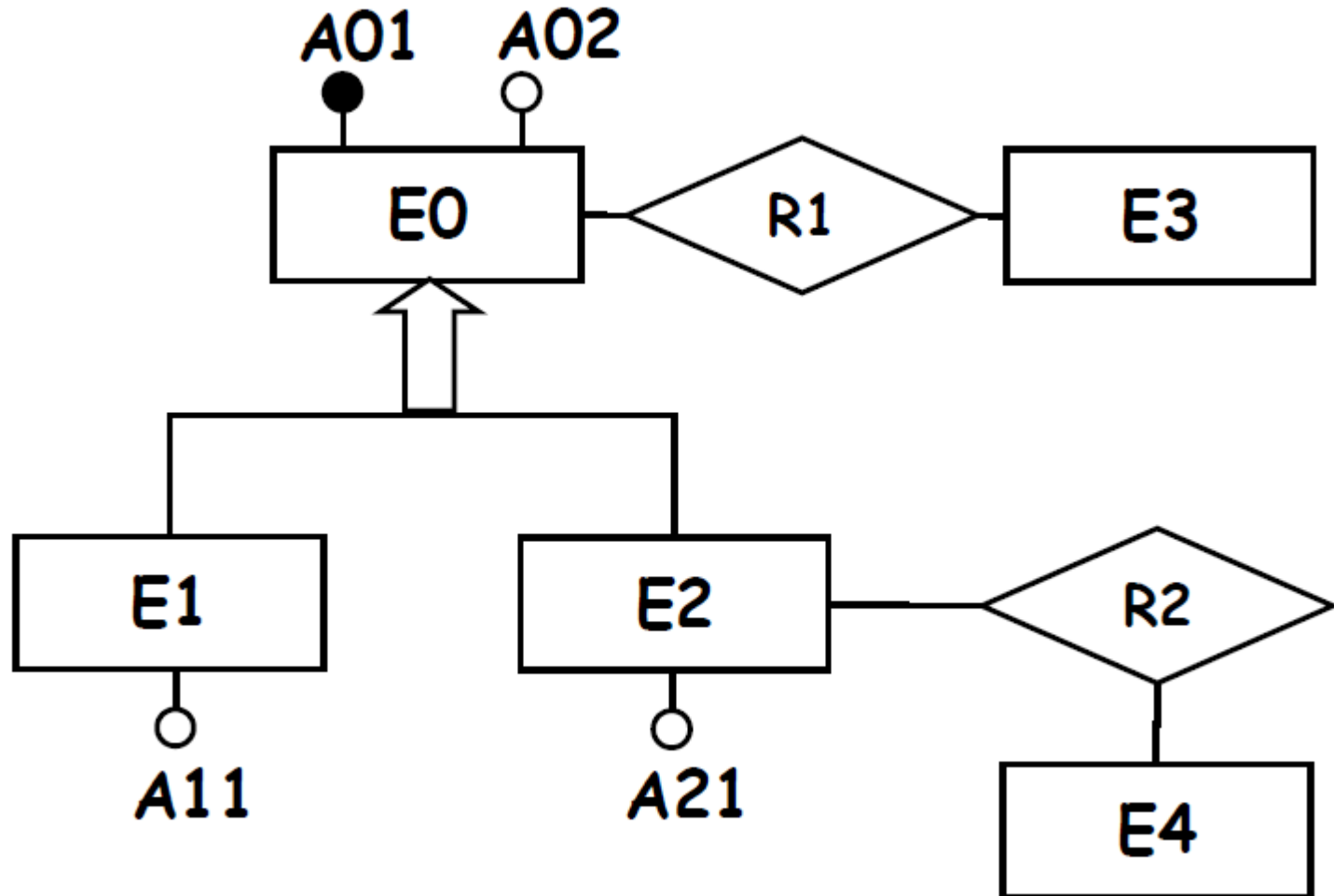
# Mantenere o no la ridondanza?

- ❑ È importante considerare la frequenza delle operazioni:
- ❑ con ridondanza:
  - operazione 1: 1500 accessi in scrittura e 500 accessi in lettura al giorno;
  - operazione 2: 2 accessi in lettura al giorno;
  - totale: 3502 accessi al giorno;
- ❑ senza ridondanza:
  - operazione 1: 1000 accessi in scrittura al giorno;
  - operazione 2: 10002 accessi in lettura al giorno;
  - totale: 12002 accessi al giorno.
- ❑ Si decide pertanto di mantenere la ridondanza, privilegiando l'efficienza.
- ❑ In generale si devono fare anche considerazioni sullo spazio in più richiesto per mantenere la ridondanza

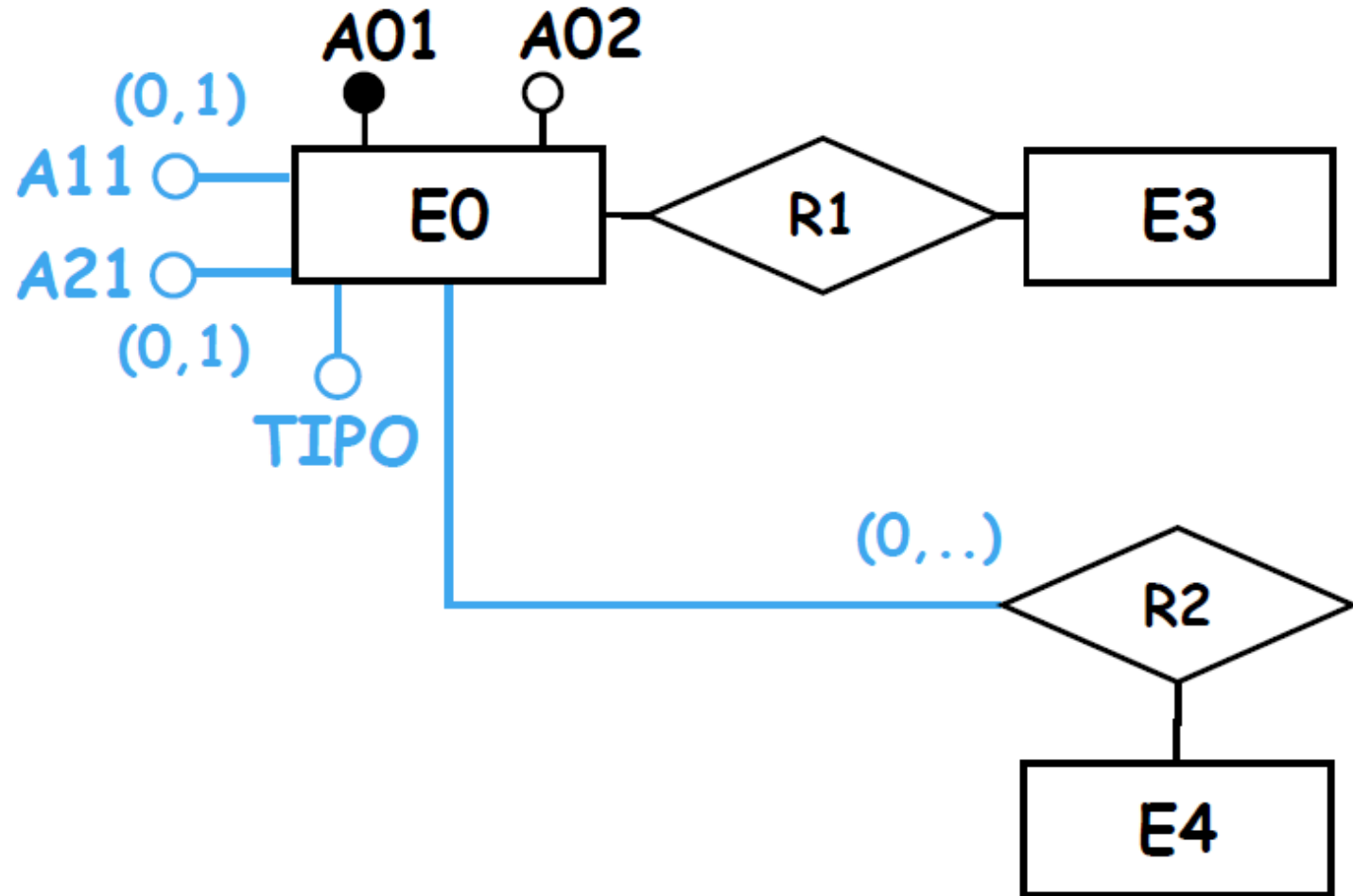
# Eliminazione delle gerarchie

- ❑ Il modello relazionale non può rappresentare direttamente le generalizzazioni.
- ❑ Entità e associazioni sono invece direttamente rappresentabili.
- ❑ Si eliminano perciò le gerarchie, sostituendole con entità e relazioni.
- ❑ Vi sono **3 possibilità** (più altre soluzioni intermedie):
  - accorpare le entità figlie nel genitore (collasso verso l'alto);
  - accorpare il genitore nelle entità figlie (collasso verso il basso);
  - sostituire la generalizzazione con associazioni.

# Schema di riferimento

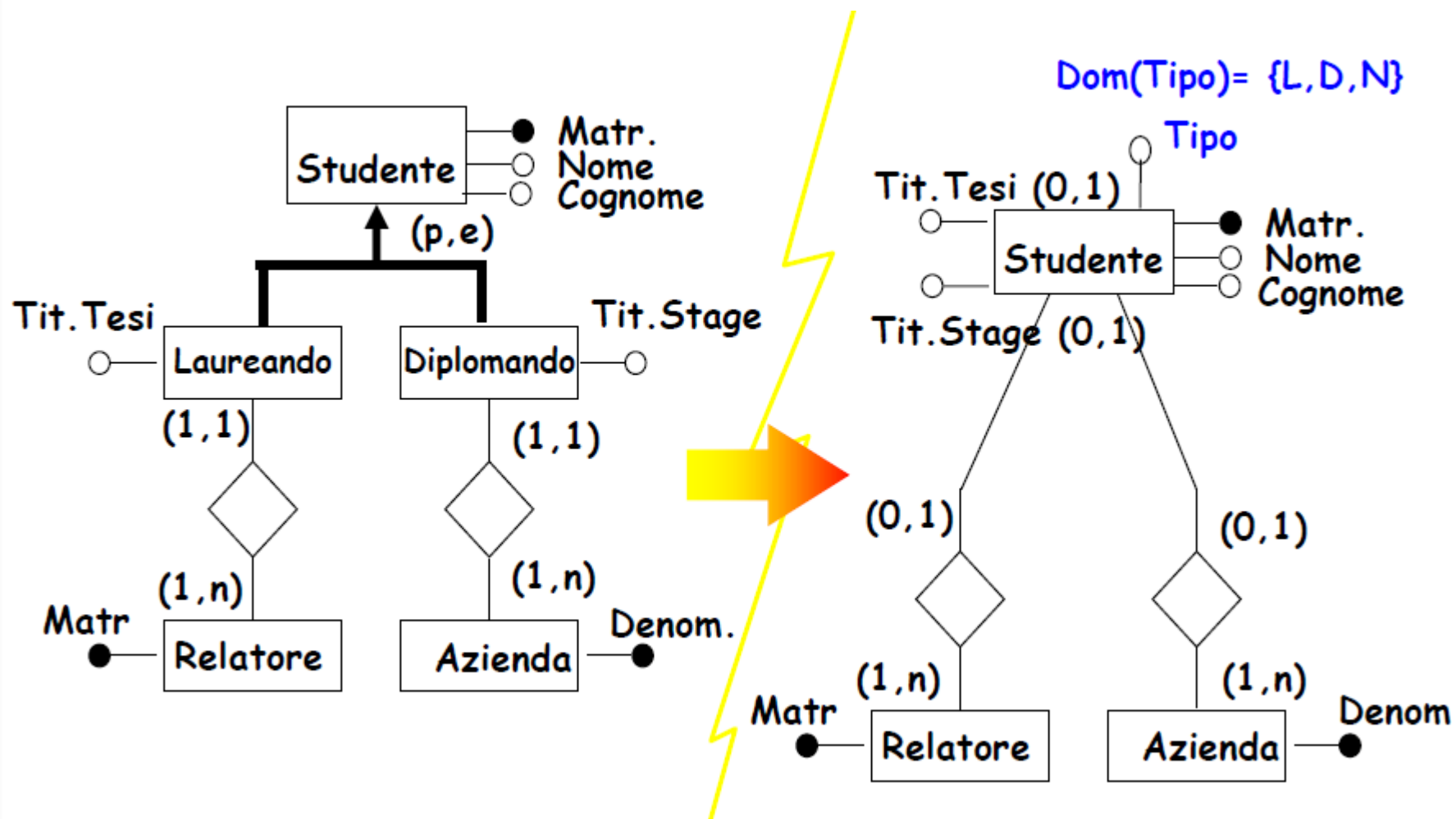


# 1. Collasso verso l'alto...



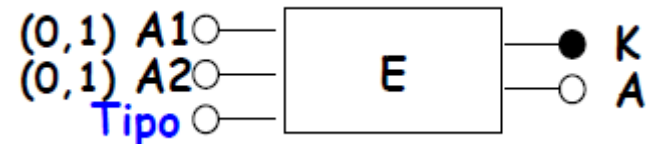


# 1. Esempio



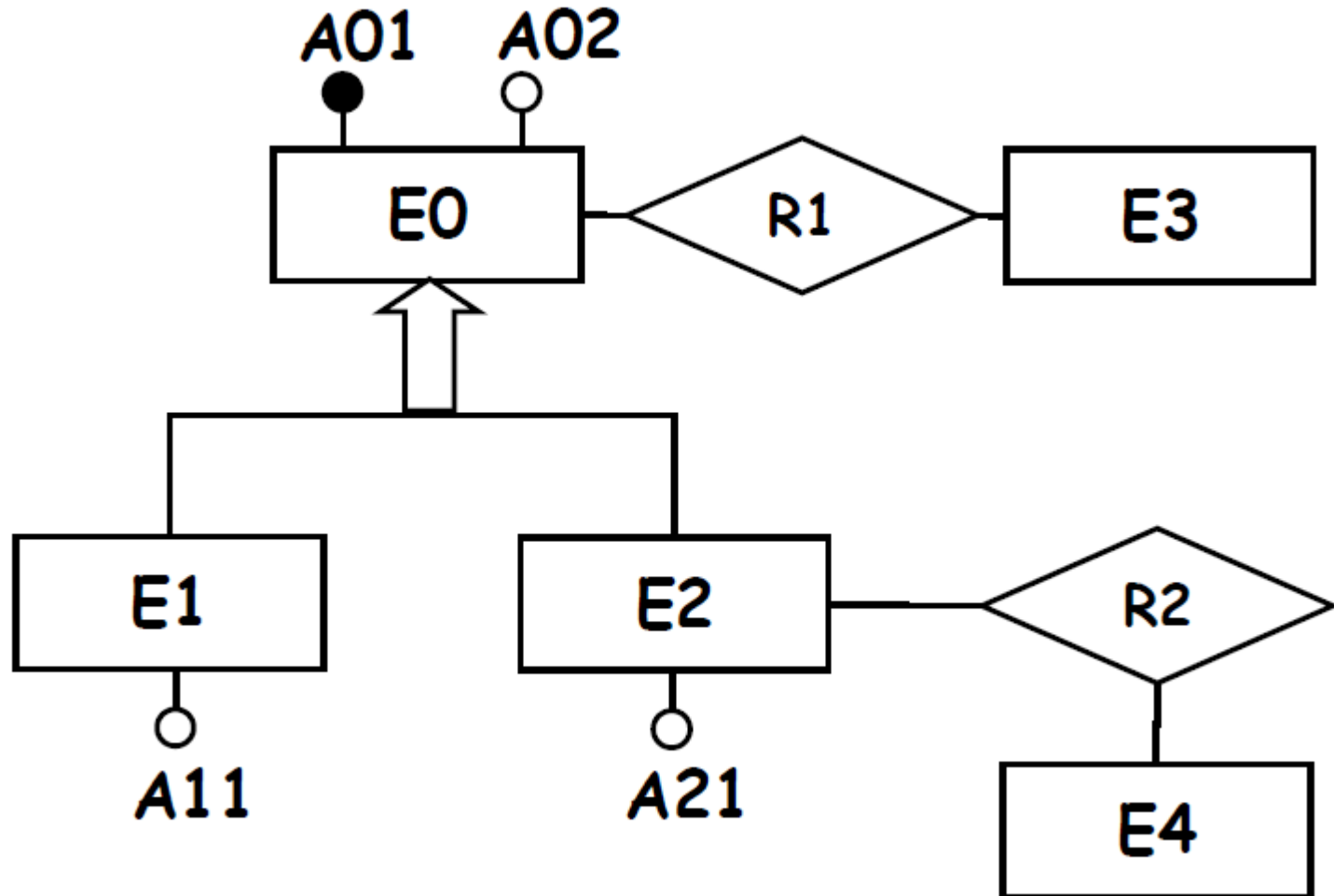
# Collasso verso l'alto: osservazioni

- “Tipo” è un attributo **selettore** che specifica se una singola istanza di E appartiene a una delle N sotto-entità.

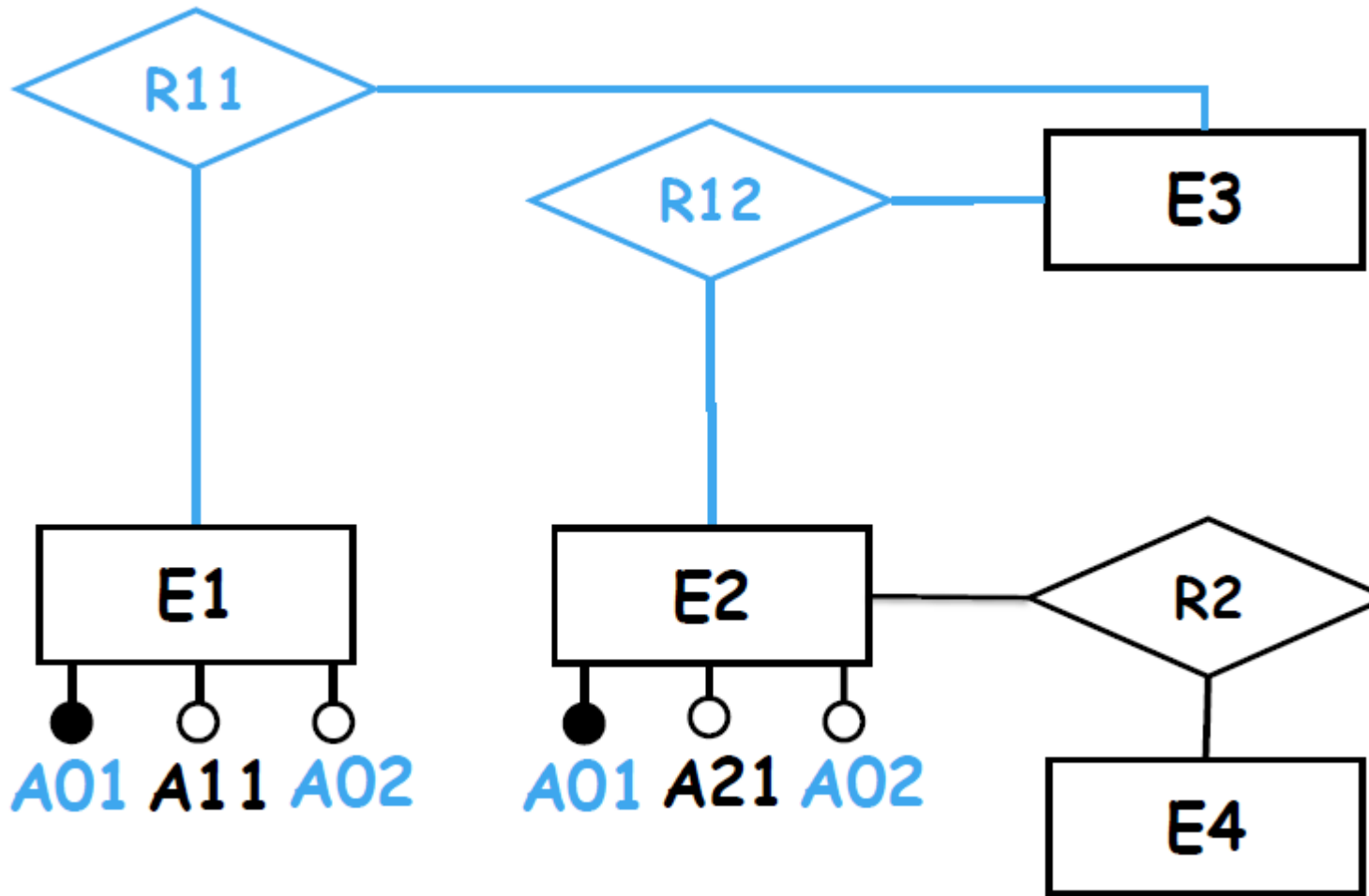


- **Copertura**
  - **totale esclusiva**: Tipo assume N valori, quante sono le sotto-entità;
  - **parziale esclusiva**: Tipo assume N+1 valori; il valore in più serve per le istanze che non appartengono a nessuna sotto-entità;
  - **sovrapposta**: occorrono tanti selettori quante sono le sotto-entità, ciascuno a valore booleano Tipo\_i, che è vero per ogni istanza di E che appartiene a E\_i; se la copertura è parziale i selettori possono essere tutti falsi, oppure si può aggiungere un selettore.
- Le eventuali associazioni connesse alle sotto-entità si trasportano su E, le eventuali cardinalità minime diventano 0.

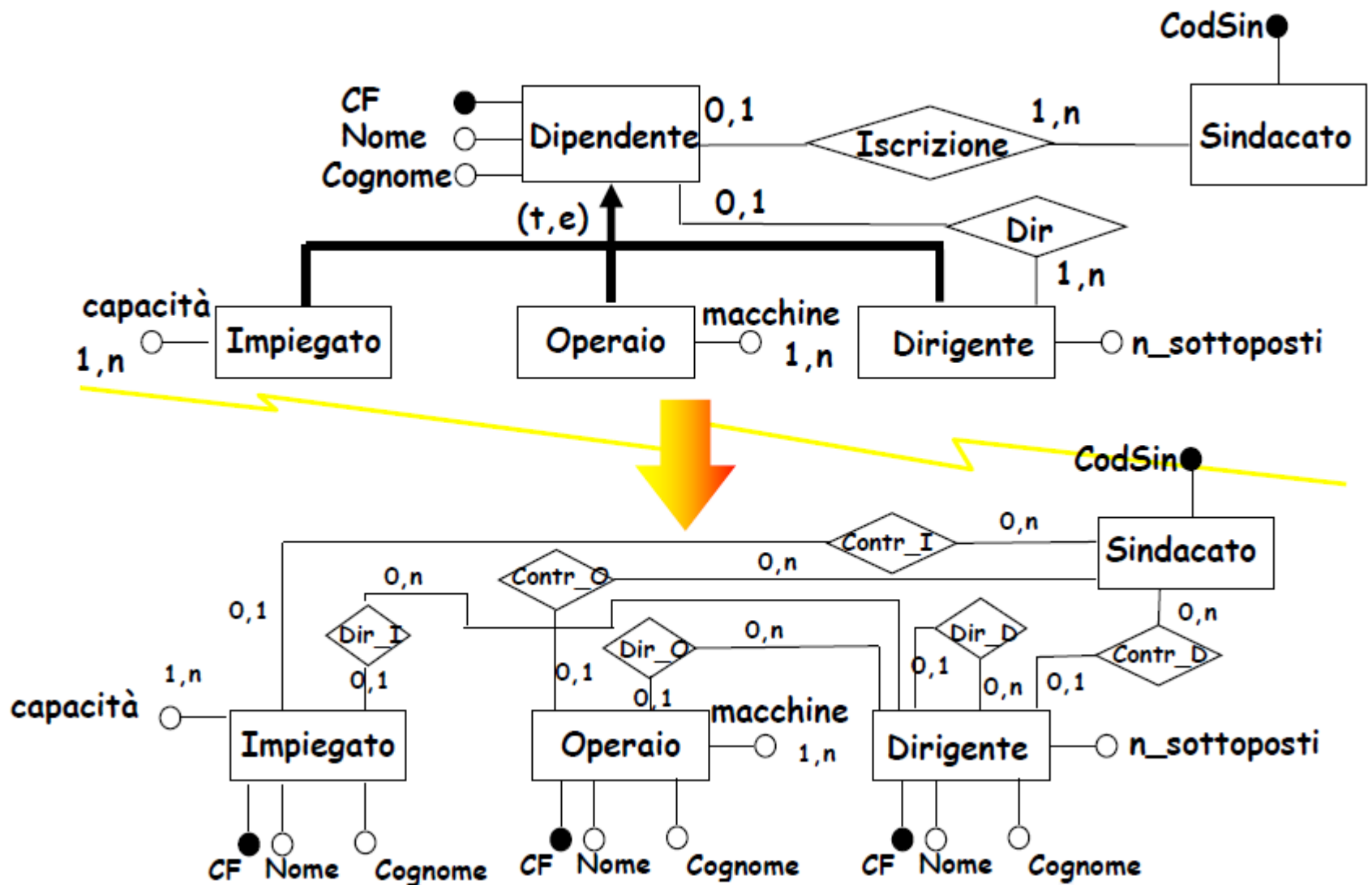
# Schema di riferimento



## 2. Collasso verso il basso...

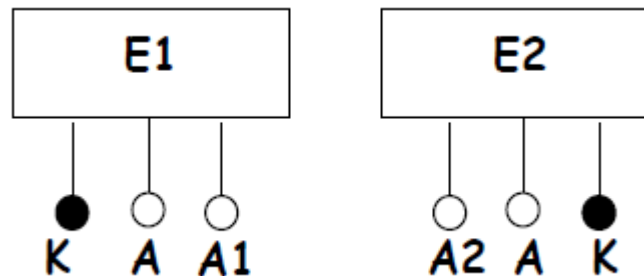


## 2. Esempio

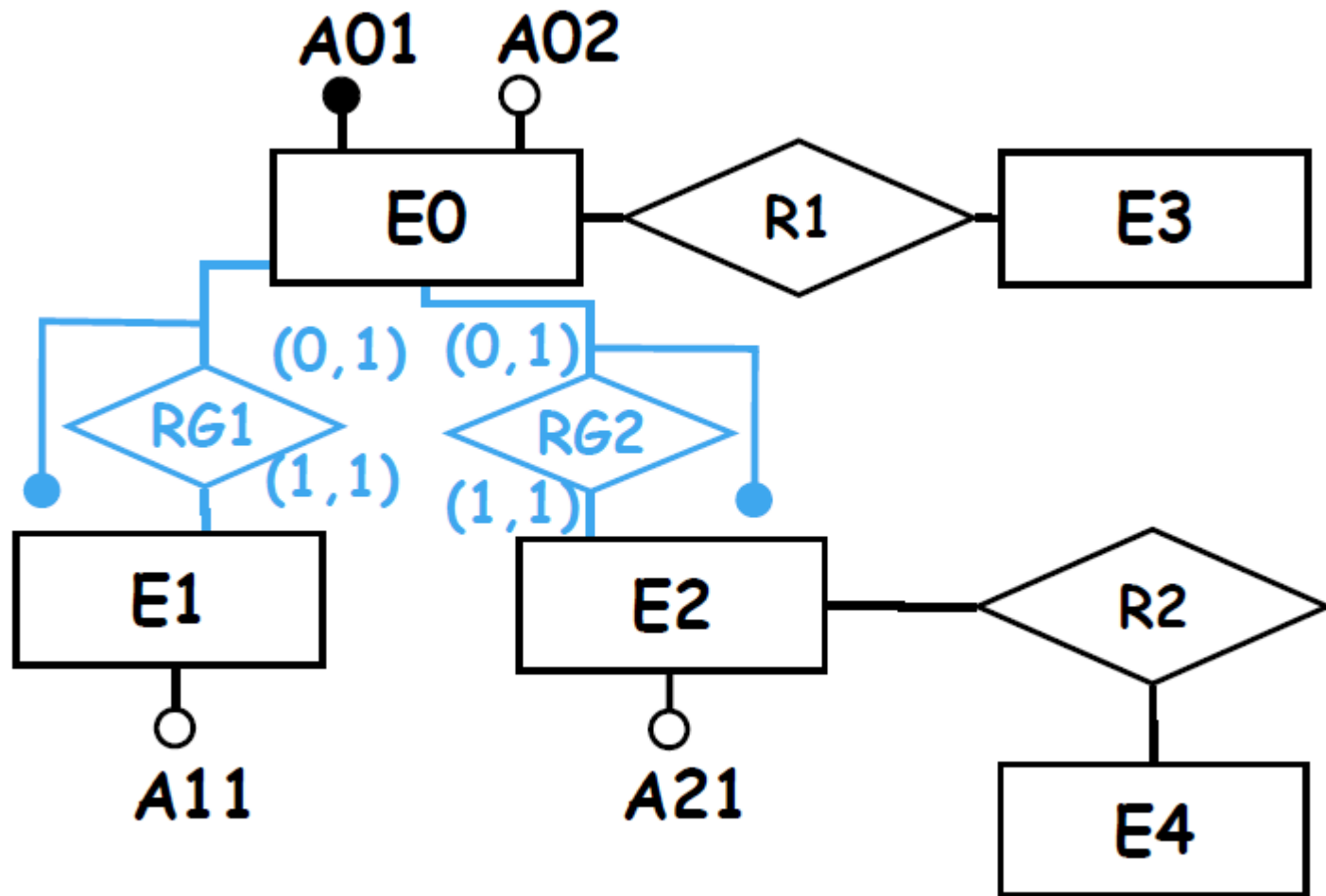


# Collasso verso il basso: osservazioni

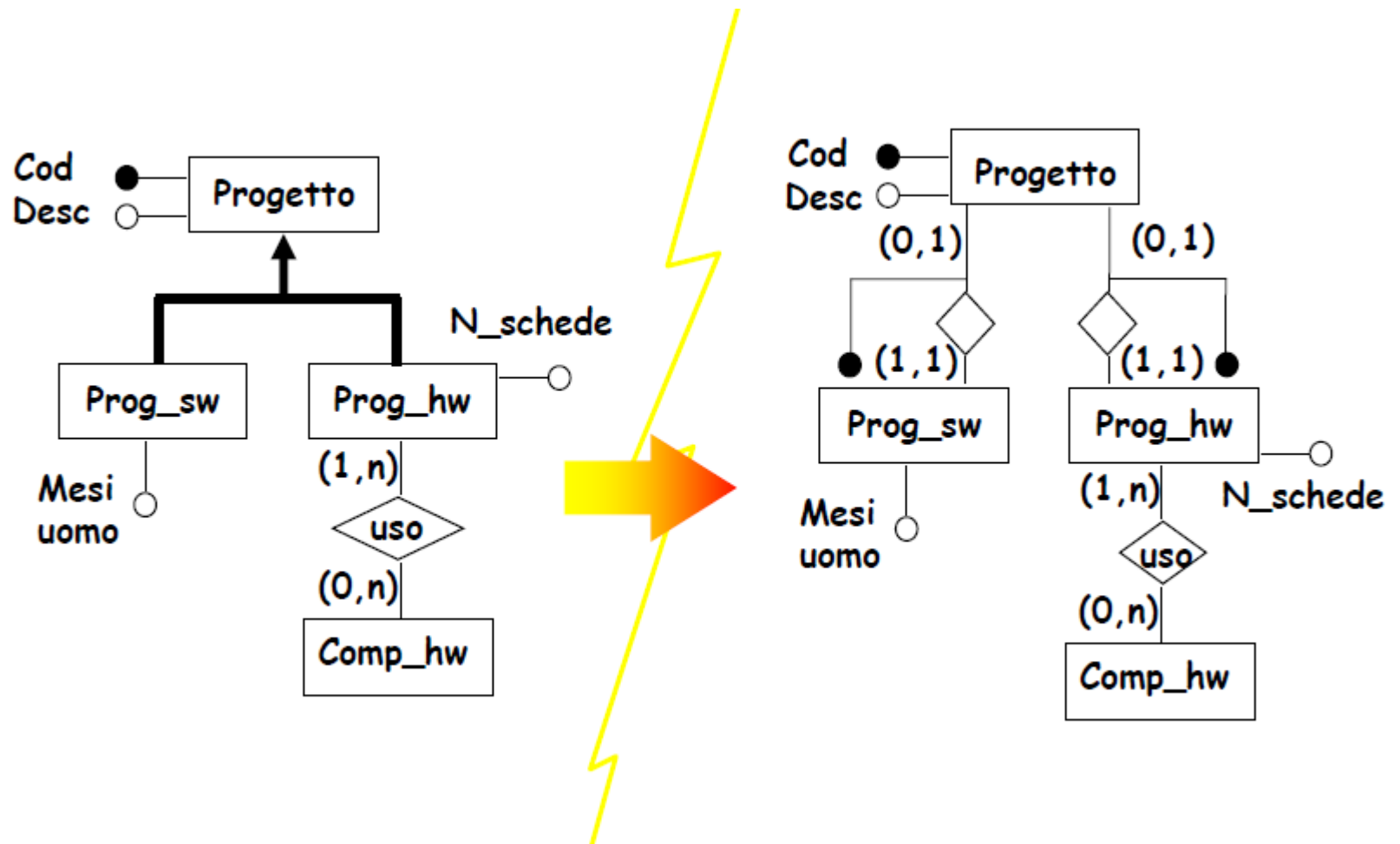
- ❑ Se la copertura non è completa, il collasso verso il basso non si può applicare:
  - non si saprebbe infatti dove collocare le istanze di E che non sono né in E1, né in E2.
- ❑ Se la copertura non è esclusiva, si introduce ridondanza:
  - una certa istanza può essere sia in E1 sia in E2, e quindi si rappresentano due volte gli attributi che provengono da E.



### 3. Sostituire con associazioni...



### 3. Esempio







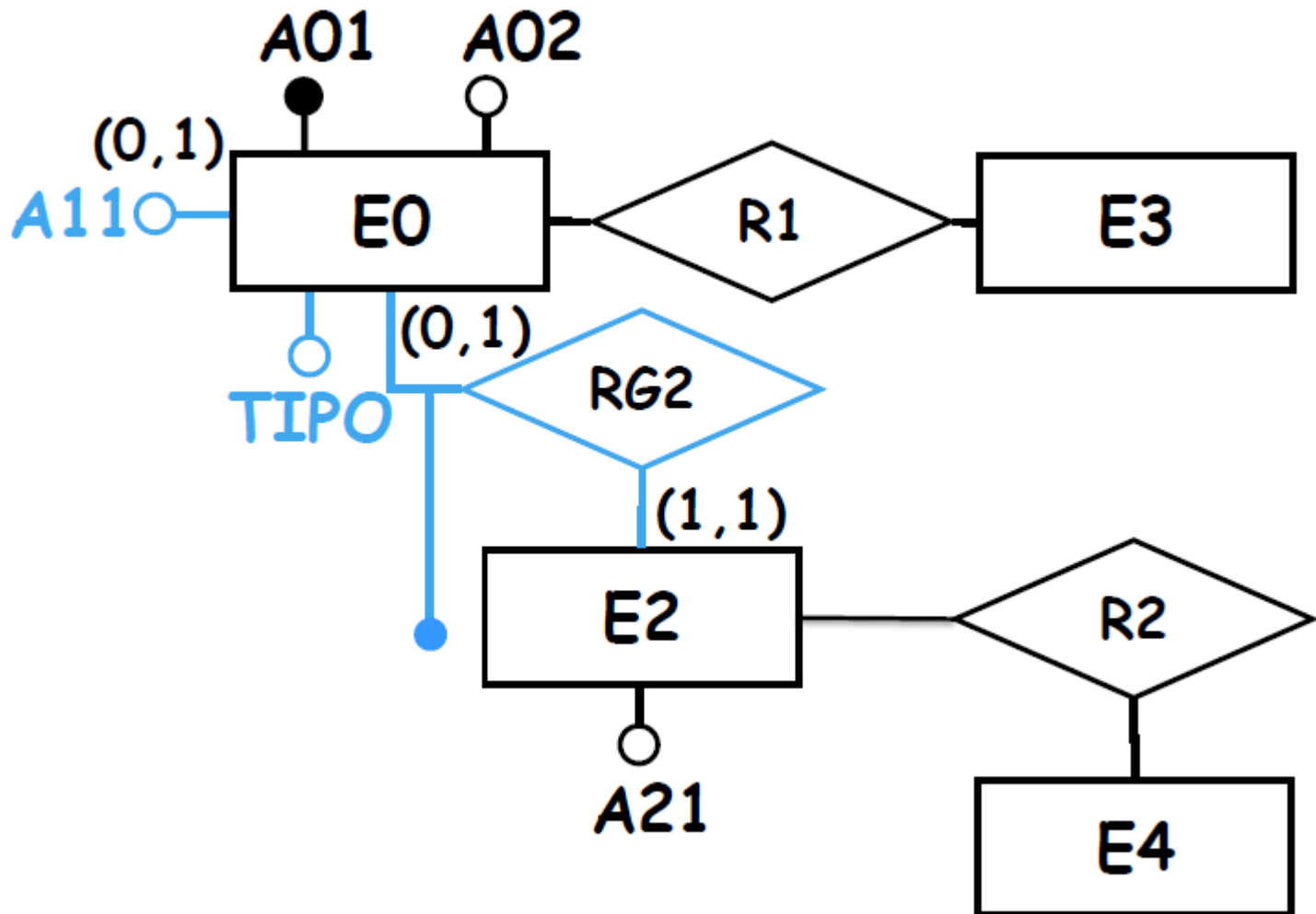
# Sostituire con associazioni: osservazioni

- ❑ Tutte le entità vengono mantenute: le entità **figlie** sono in associazione binaria con l'entità **padre** e sono **identificate esternamente**.
- ❑ La sostituzione con associazioni è sempre possibile, indipendentemente dalla copertura della gerarchia.

# Quale alternativa scegliere?

- ❑ La scelta fra le alternative illustrate si può fare adottando un **metodo simile a quello visto per l'analisi delle ridondanze**, considerando sia il numero degli accessi sia l'occupazione di spazio.
- ❑ È possibile seguire alcune **semplici regole generali** (ovvero: **mantieni insieme ciò che viene usato insieme**):
  1. conviene se gli accessi all'entità padre e alle entità figlie sono contestuali;
  2. conviene se gli accessi alle entità figlie sono distinti, ma d'altra parte è possibile solo con generalizzazioni totali;
  3. conviene se gli accessi alle entità figlie sono separati dagli accessi al padre.
- ❑ Sono anche possibili soluzioni **“ibride”**, soprattutto in presenza di gerarchie a più livelli.

# Una soluzione ibrida...

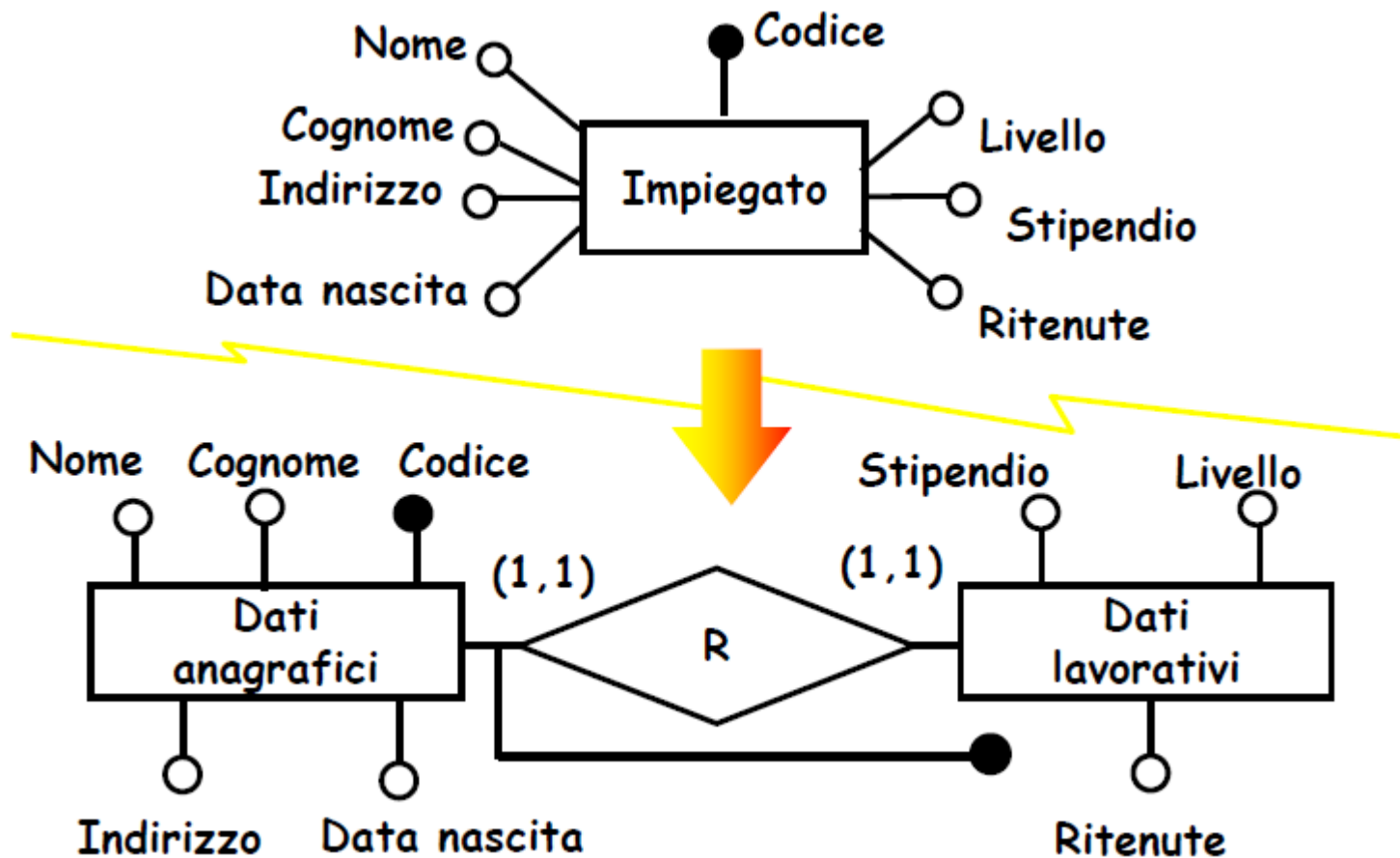


# Partizionamenti e accorpamenti

- ❑ È possibile ristrutturare lo schema accorpendo o partizionando entità e associazioni.
- ❑ Tali ristrutturazioni sono effettuate per rendere più efficienti le operazioni in base al principio già visto, ovvero:
- ❑ **gli accessi si riducono:**
  - separando attributi di un concetto che vengono acceduti separatamente;
  - raggruppando attributi di concetti diversi a cui si accede insieme.
- ❑ I casi principali sono:
  - **partizionamento verticale di entità;**
  - **partizionamento orizzontale di associazioni;**
  - **eliminazione di attributi multivalore;**
  - **accorpamenti di entità e associazioni.**

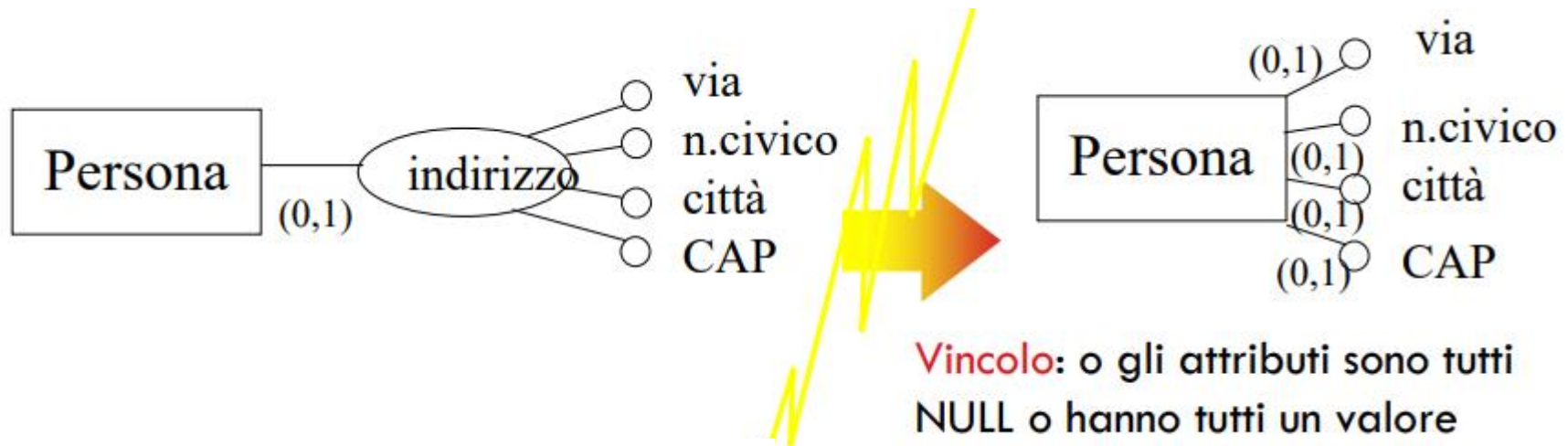
# Partizionamento verticale di entità

- Si separano gli attributi in gruppi omogenei:



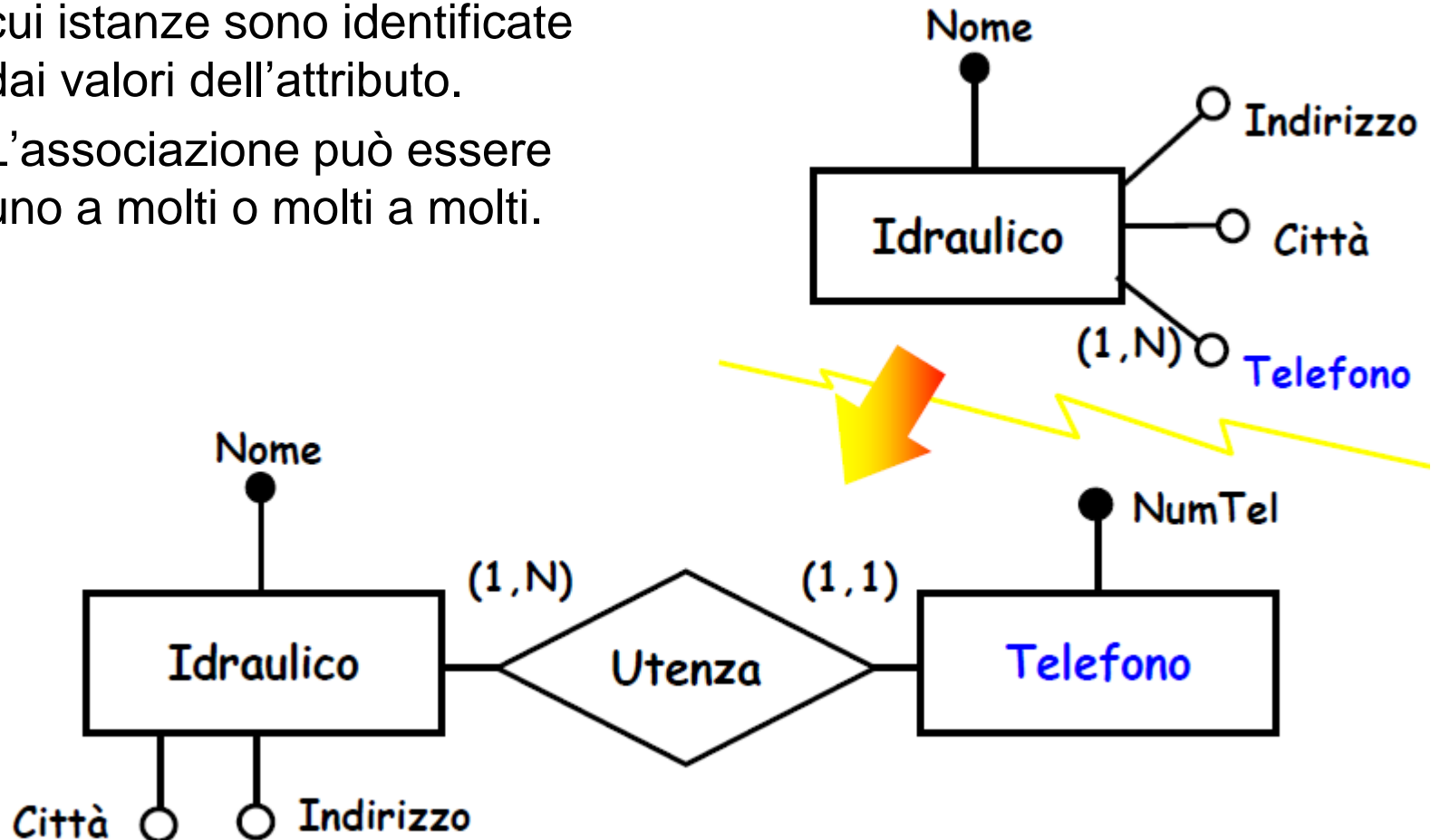
# Eliminazione di attributi composti e/o multivalore

- ❑ Il vincolo che i domini siano atomici rende attributi **composti** (record) e **multivalore** (array) non mappabili nel modello relazionale
- ❑ Gli attributi composti vengono semplicemente scomposti in attributi semplici (**introducendo dei vincoli per gestire eventuali opzionalità**)



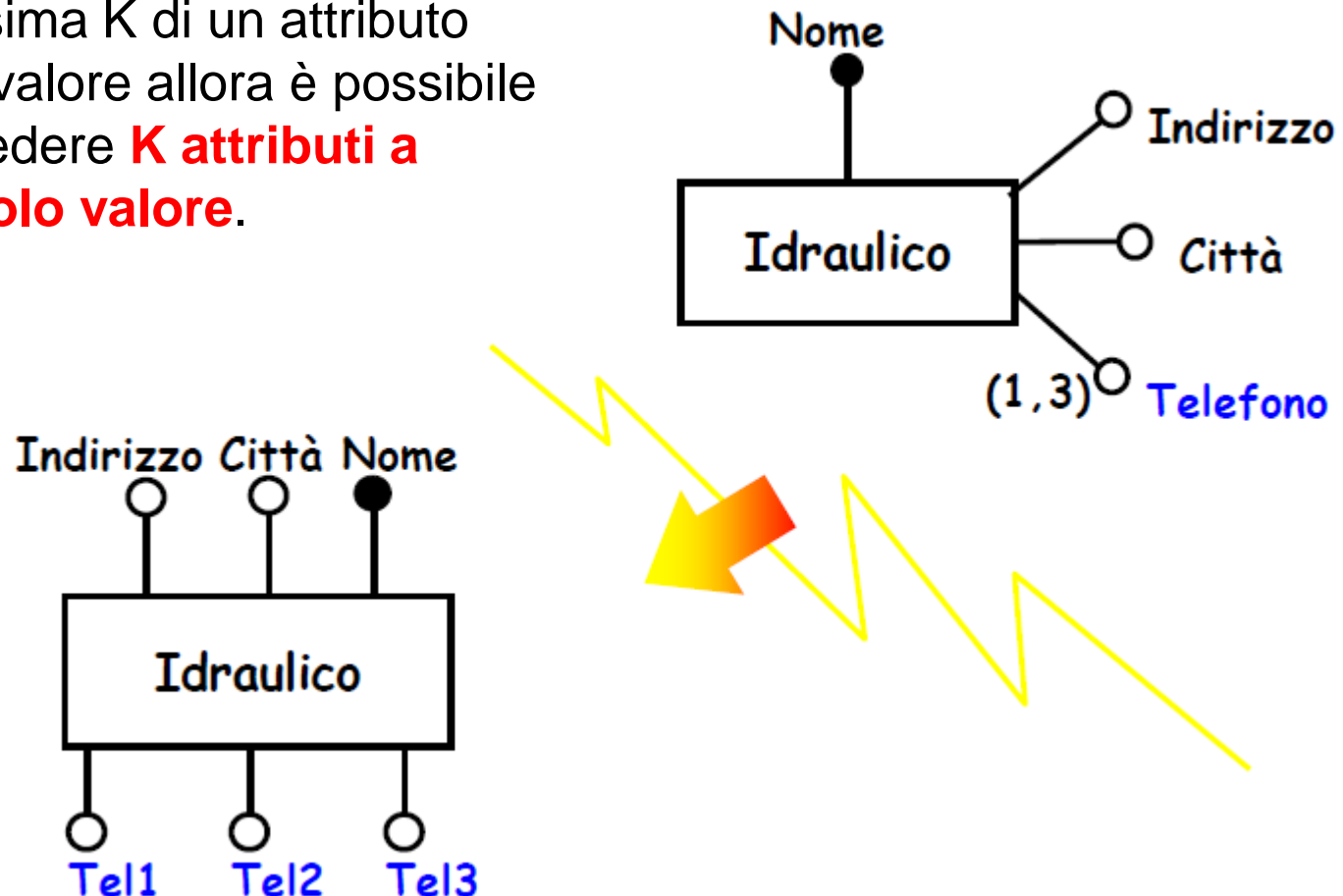
# Eliminazione di attributi multivalore (1)

- Si introduce una **nuova entità** le cui istanze sono identificate dai valori dell'attributo.
- L'associazione può essere uno a molti o molti a molti.



# Eliminazione di attributi multivalore (2)

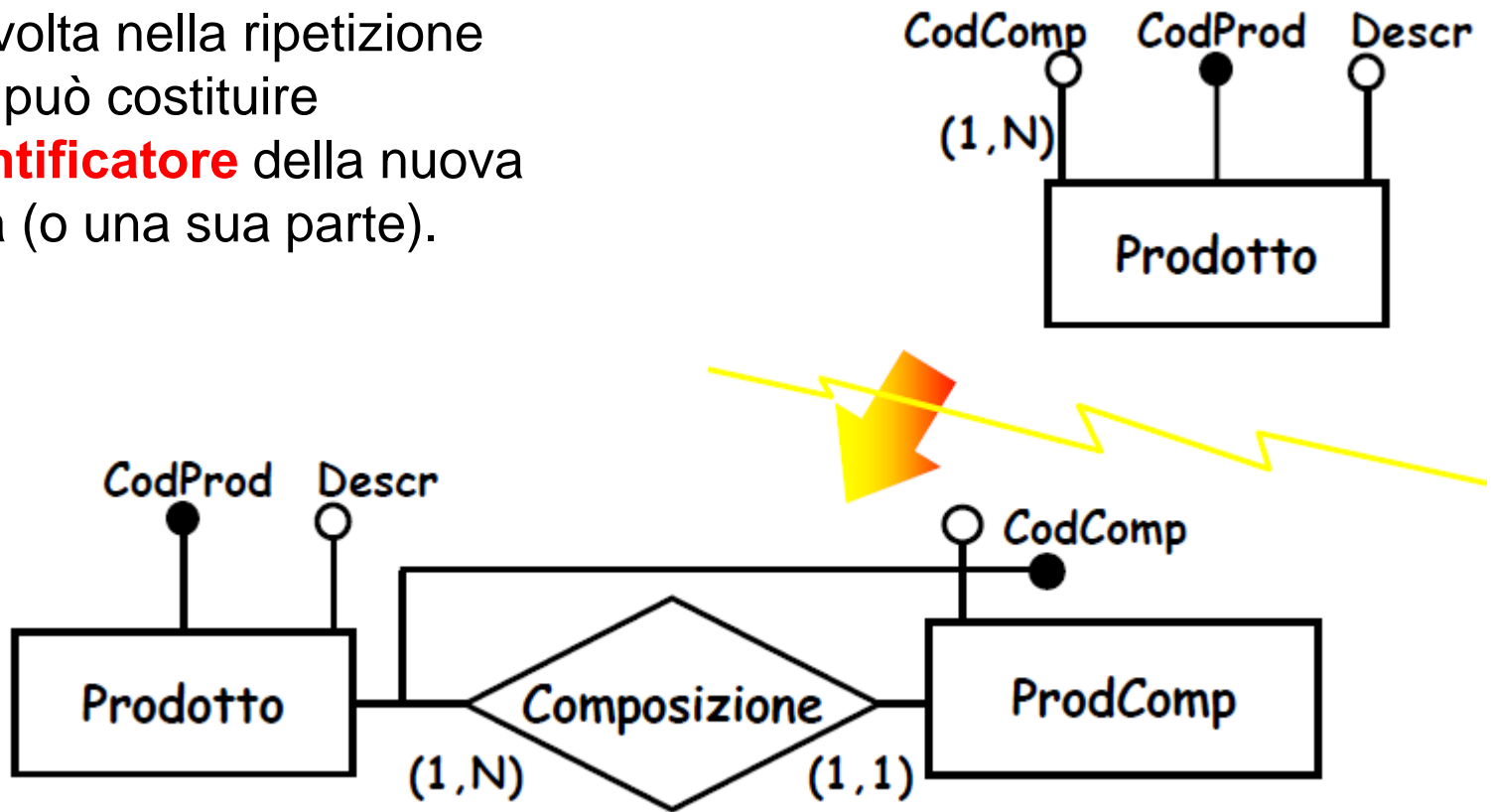
- Se è nota la cardinalità massima  $K$  di un attributo multivalore allora è possibile prevedere  **$K$  attributi a singolo valore**.





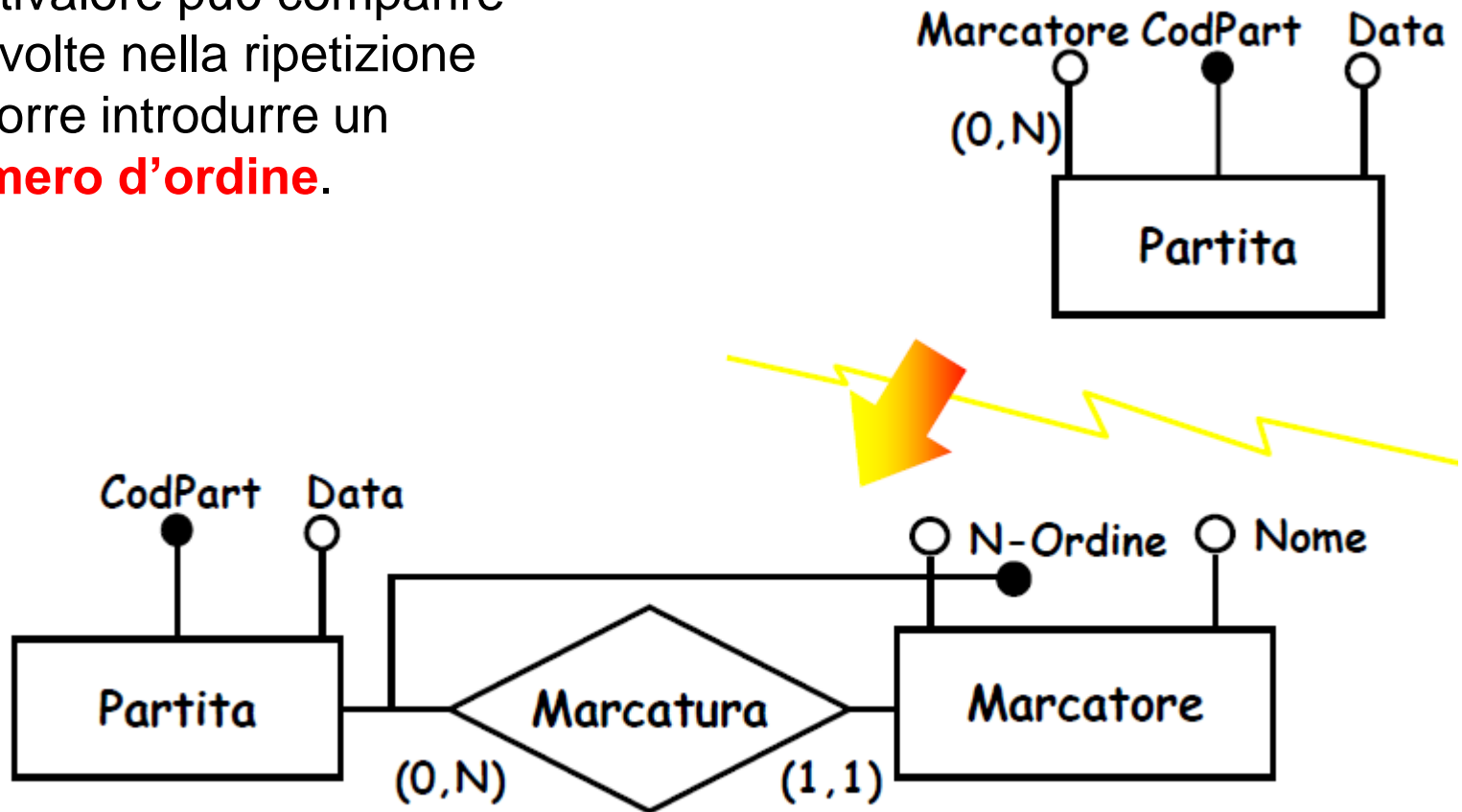
# Eliminazione di attributi multivalore (3)

- Se un valore dell'attributo multivalore compare una sola volta nella ripetizione esso può costituire **l'identificatore** della nuova entità (o una sua parte).



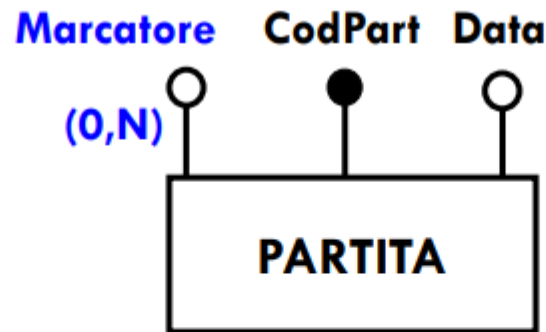
# Eliminazione di attributi multivalore (4)

- Se un valore dell'attributo multivalore può comparire più volte nella ripetizione occorre introdurre un **numero d'ordine**.



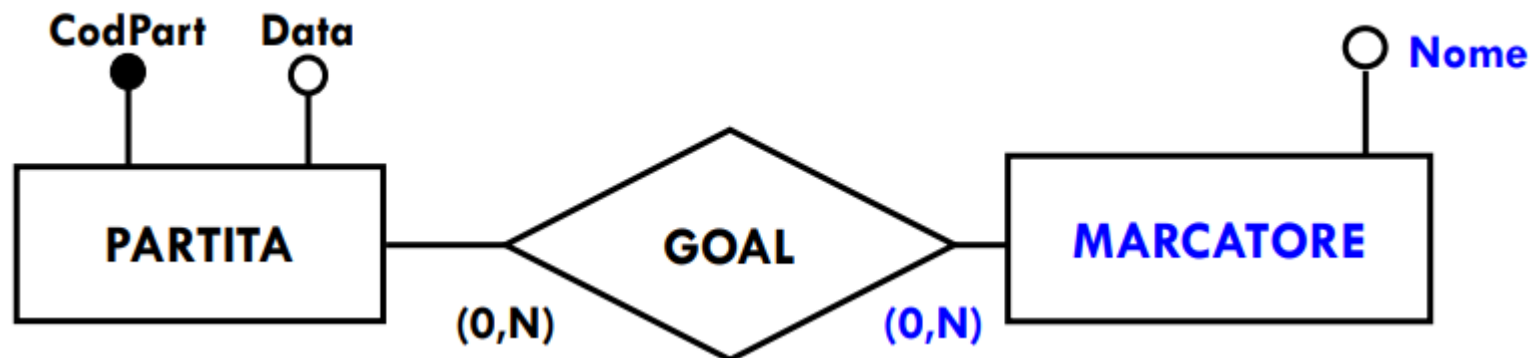
# Esercizio 3

- ❑ Eliminare l'attributo multivalore

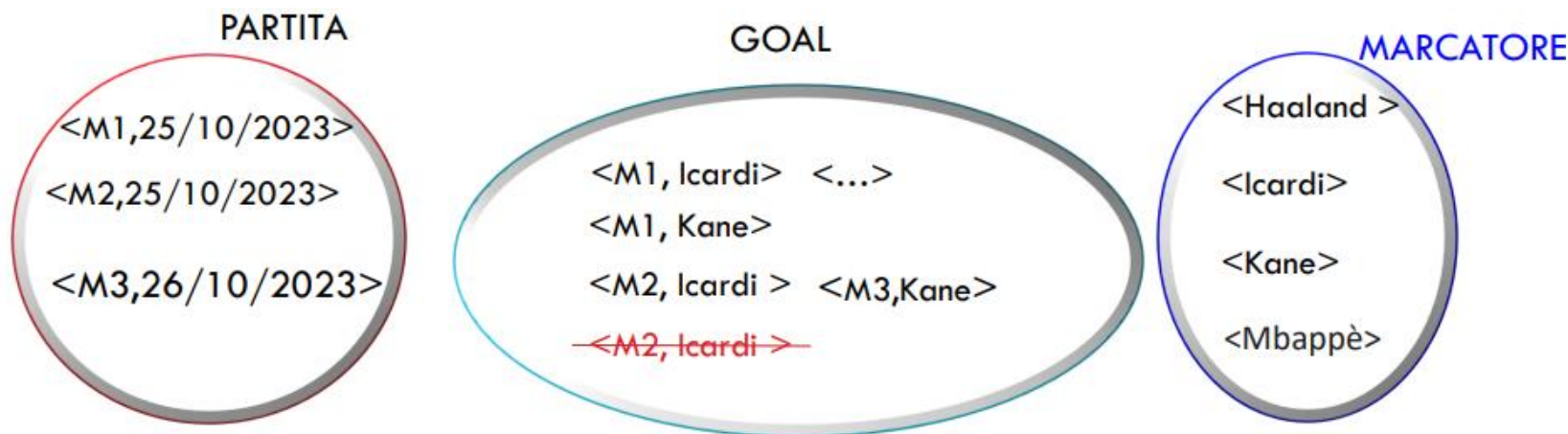


Nota: le istanze di marcatore possono essere ripetute: un marcatore può segnare 2 gol in partite diverse o anche nella stessa partita

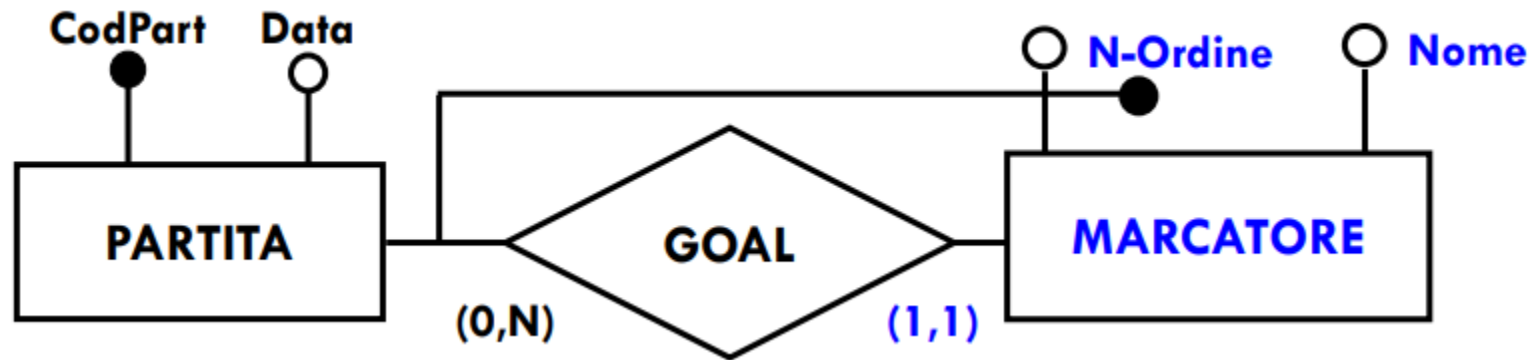
## Esercizio 3 - soluzione



- ❑ Questa soluzione non è corretta perché un marcatore non potrebbe segnare più di un goal nella stessa partita



# Esercizio 3 - soluzione

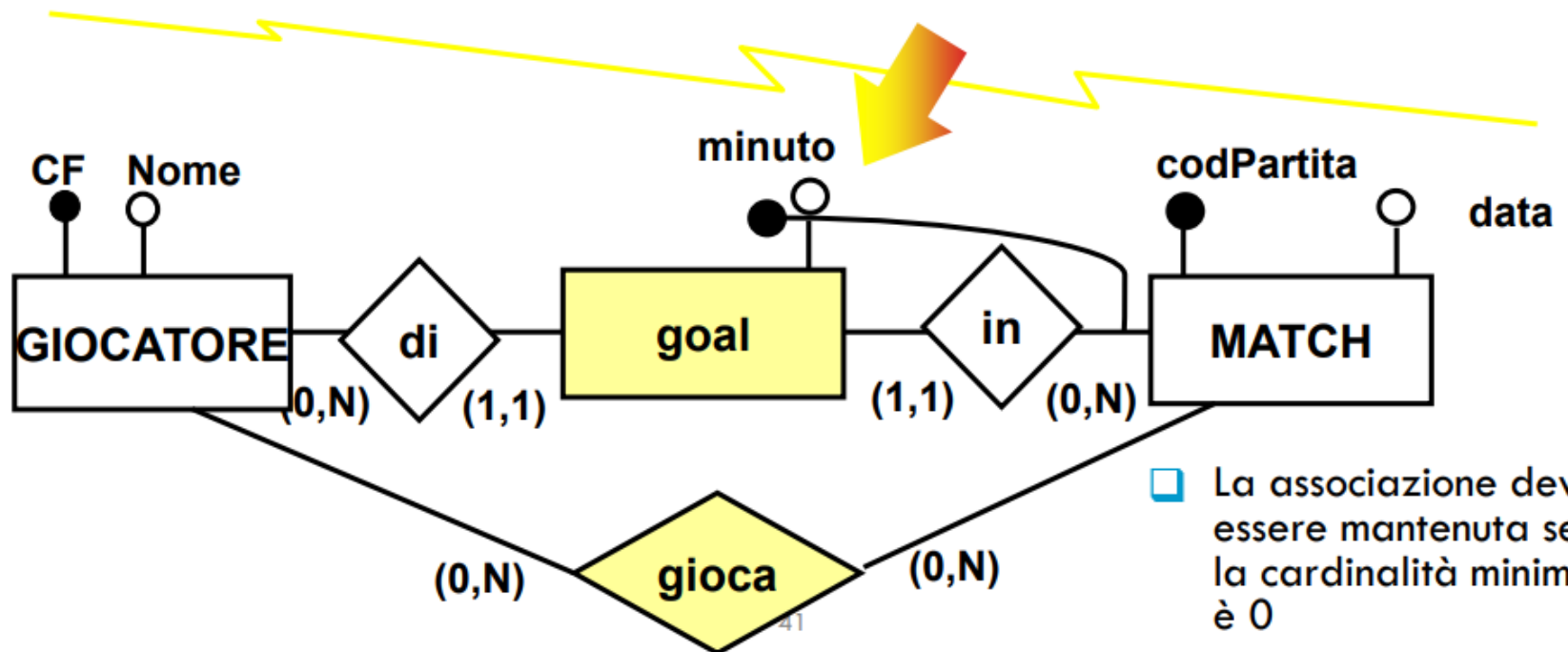
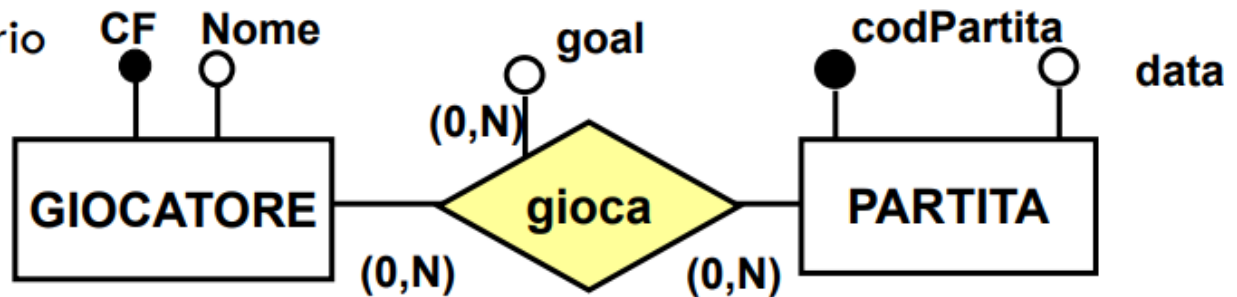


- Se un valore dell'attributo multivalore può comparire più volte nella ripetizione si può introdurre un numero d'ordine



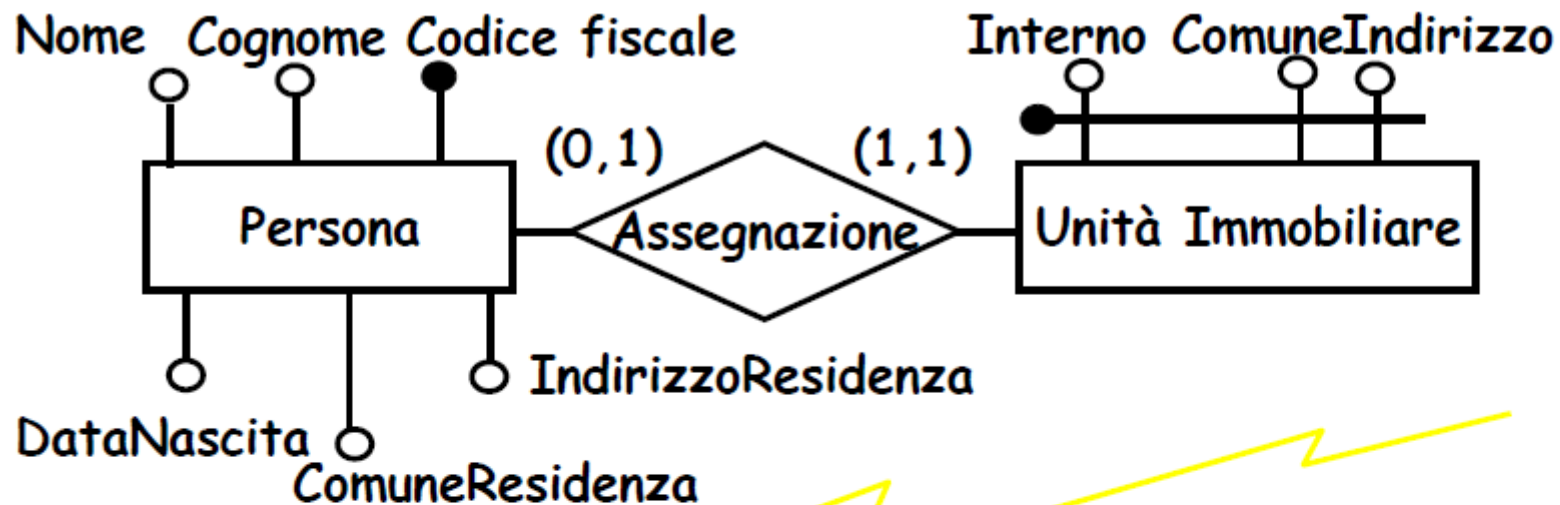
# Eliminazione di attributi multivalore nelle associazioni

- È prima necessario reificare la associazione in un'entità

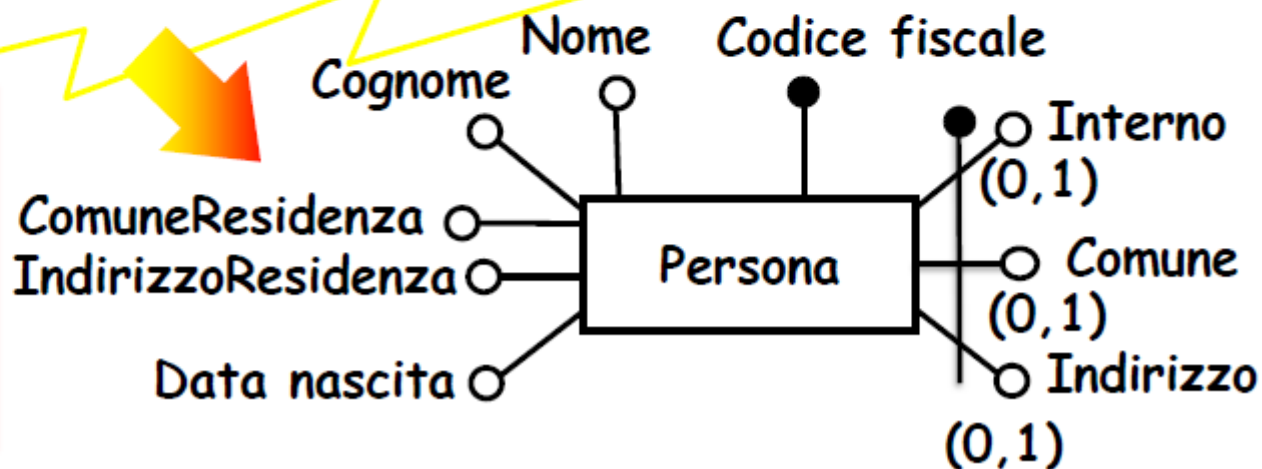


- La associazione deve essere mantenuta se la cardinalità minima è 0

# Accorpamento di entità



Questa soluzione è sensata se e solo se non si ha interesse a mantenere, nemmeno per evoluzioni future dello schema, l'entità Unità Immobiliare.



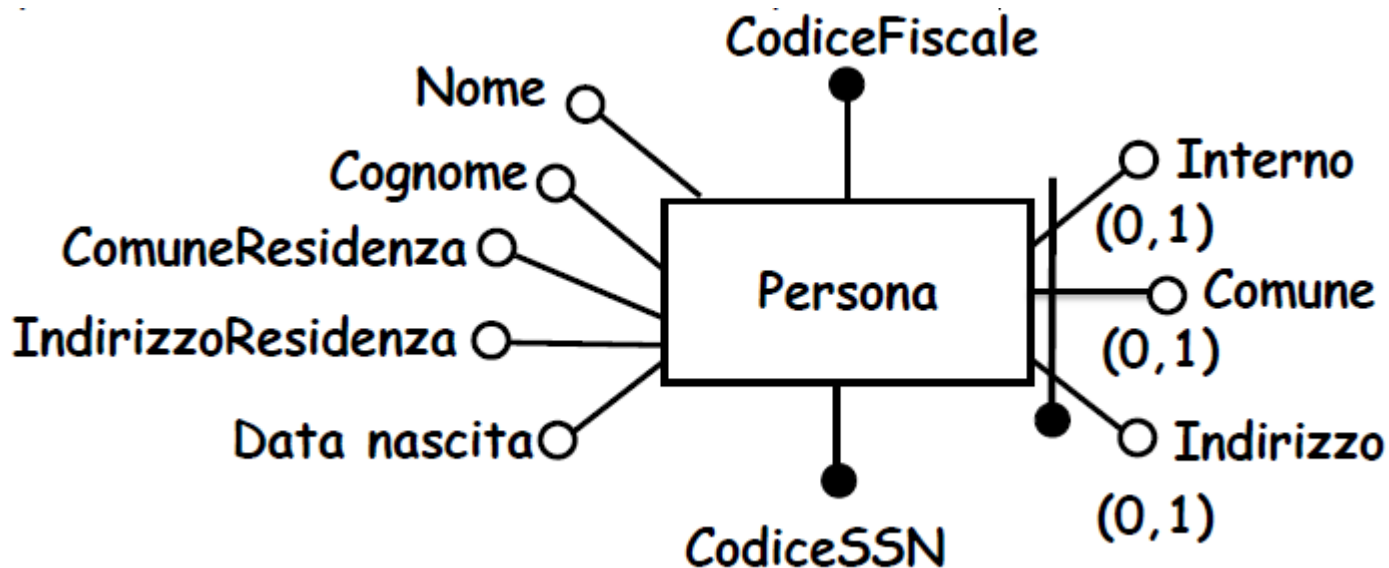
# Scelta degli identificatori principali

- ❑ È un'operazione indispensabile per la traduzione nel modello relazionale, e corrisponde alla scelta della **chiave primaria**.
- ❑ I criteri da adottare sono:
  - assenza di opzionalità (valori NULL);
  - semplicità;
  - utilizzo nelle operazioni più frequenti o importanti.
- ❑ Se nessuno degli identificatori soddisfa i requisiti si introducono nuovi attributi (**codici**) ad hoc.



# Identificatori principali: esempio

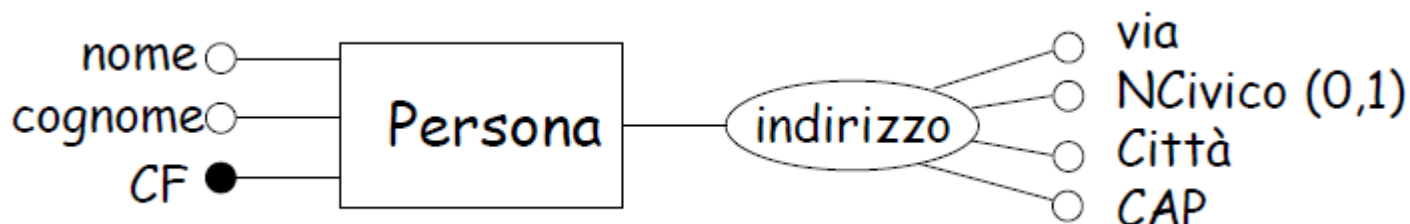
- ❑ L'identificatore {Interno, Comune, Indirizzo} è opzionale, quindi non può essere scelto come chiave primaria.
- ❑ Tra gli attributi CodiceFiscale e CodiceSSN la scelta dipende da quale fra questi è più frequentemente usato per accedere ai dati di una persona.



# Traduzione delle entità

## Idea di base:

- ❑ Ogni entità è tradotta con una relazione con gli stessi attributi.
  - La **chiave primaria** coincide con l'**identificatore principale** dell'entità.
  - Gli **attributi composti** vengono ricorsivamente suddivisi nelle loro componenti, oppure sono mappati in un singolo attributo della relazione, il cui dominio deve essere opportunamente definito.
  - Si usa l'asterisco (\*) per indicare la possibilità di **valori nulli**.



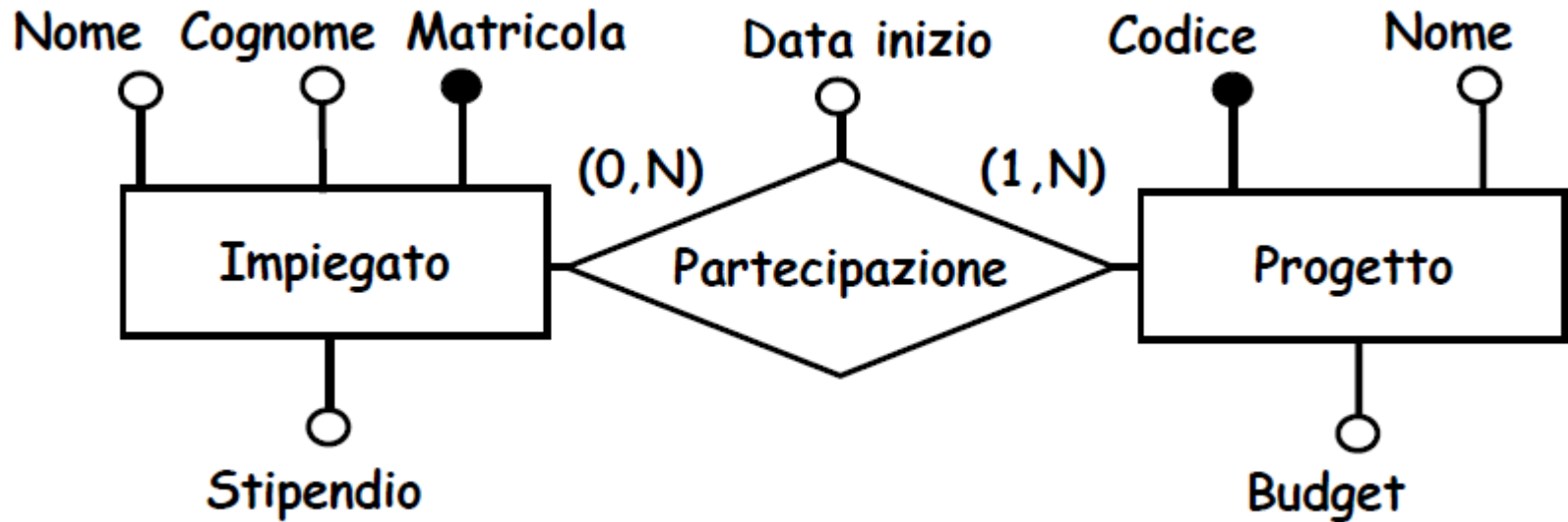
- ❑ **Persona** (CF, Cognome, Nome, Via, NCivico\*, Città, CAP)

# Traduzione delle associazioni

## Idea di base:

- ❑ Ogni associazione è tradotta con una relazione con gli stessi attributi, cui si aggiungono gli identificatori di tutte le entità che essa collega.
  - Gli **identificatori delle entità** collegate costituiscono una **superchiave**.
  - La **chiave** dipende dalle **cardinalità massime** delle entità nell'associazione.
  - Le **cardinalità minime** determinano, a seconda del tipo di traduzione effettuata, la presenza o meno di **valori nulli** (e quindi incidono sui vincoli e sull'occupazione di memoria).

# Entità e associazione molti a molti



- ❑ **Impiegato** (Matricola, Nome, Cognome, Stipendio)
- ❑ **Progetto** (Codice, Nome, Budget)
- ❑ **Partecipazione** (Matricola, Codice, DataInizio)
  - FK: Matricola REFERENCES Impiegato
  - FK: Codice REFERENCES Progetto

# Entità e associazione molti a molti

Impiegato(Matricola, Cognome, Stipendio)  
Partecipazione(Matricola, Codice, DataInizio)  
Progetto(Codice, Nome, Budget)

Impiegato

Cognome	Stipendio	<u>Matricola</u>
Rossi	500	0801
Verdi	800	0802
Bianchi	400	0823

<u>Codice</u>	Budget	Nome
P0032	3500	Alpha
P0054	7800	Beta
P0012	2400	Gamma

Progetto

Partecipazione

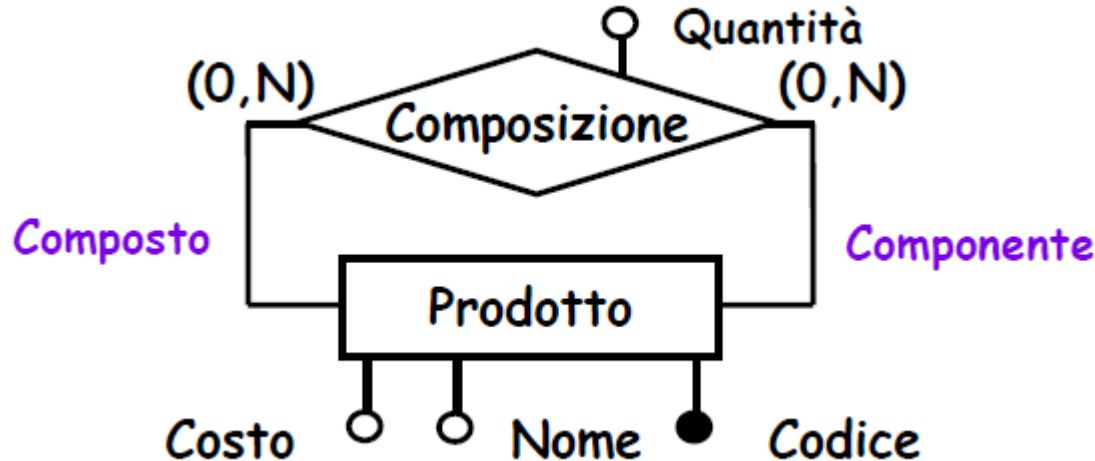
<u>Matricola</u>	DataInizio	<u>Codice</u>
0801	22/3/2001	P0032
0801	12/11/2001	P0012
0802	10/7/2001	P0054

# Nomi delle foreign key

- ❑ Non è ovviamente necessario mantenere, per gli attributi chiave della relazione che traduce l'associazione, gli stessi nomi delle primary key referenziate, conviene piuttosto far ricorso a nomi più espressivi.
- ❑ **Partecipazione** (Impiegato, CodProgetto, DataInizio)
  - FK: Impiegato REFERENCES Impiegato
  - FK: CodProgetto REFERENCES Progetto
- ❑ **Ovviamente se le entità collegate hanno un attributo con lo stesso nome la ridenominazione è obbligatoria!**

# Associazioni ad anello molti a molti

- ❑ In questo caso i nomi degli attributi che formano la chiave primaria della relazione che traduce l'associazione si possono derivare dai **ruoli** presenti nei rami dell'associazione stessa.



- ❑ **Prodotto** (Codice, Nome, Costo)
- ❑ **Composizione** (Composto, Componente, Quantità)
  - FK: Composto REFERENCES Prodotto
  - FK: Componente REFERENCES Prodotto

# Associazioni ad anello molti a molti

Prodotto(Codice, Nome, Costo)

Composizione(Composto, Componente, Quantità)

Prodotto

Nome	Costo	<u>Codice</u>
Cilindro	500	C003
Cuscinetto	800	C023
Sfera	300	V823

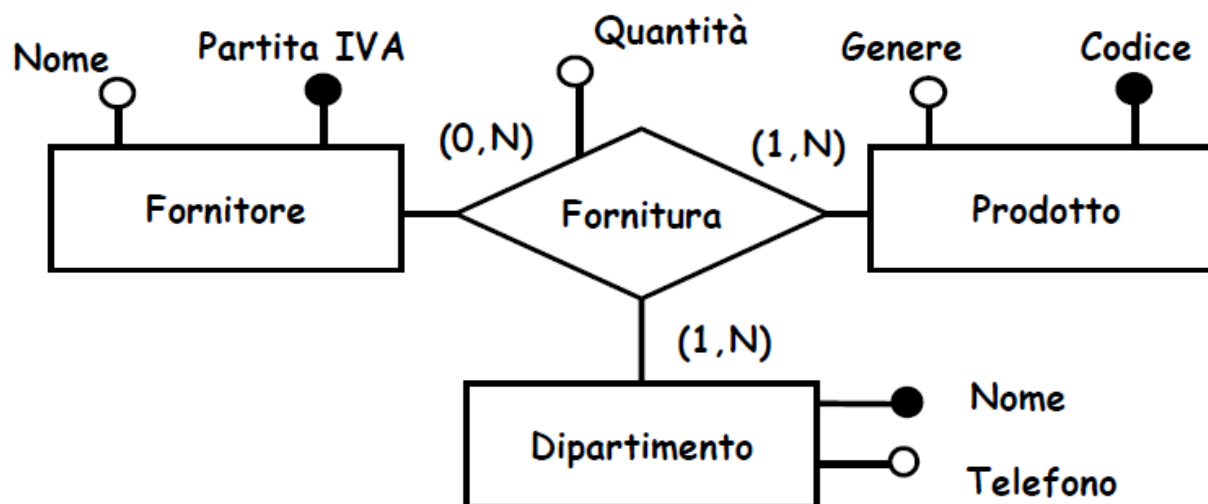
<u>Composto</u>	<u>Componente</u>	Quantità
C023	C003	2
C023	V823	20

Composizione



# Associazioni n-arie molti a molti

- ❑ La chiave è la combinazione degli identificatori delle N entità partecipanti.



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

➤ FK: ...

# Associazioni n-arie molti a molti

Fornitore(PartitaIVA, Nome)

Prodotto(Codice, Genere), Dipartimento(Nome, Telefono)

Fornitura(Fornitore, Prodotto, Dipartimento, Quantità)

Nome	<u>PartitaIVA</u>
Rossi	08009382
Verdi	07092913
Bianchi	04563281

Fornitore

<u>Codice</u>	Genere
A0034	Computer
B3456	Stampante
V0567	Video

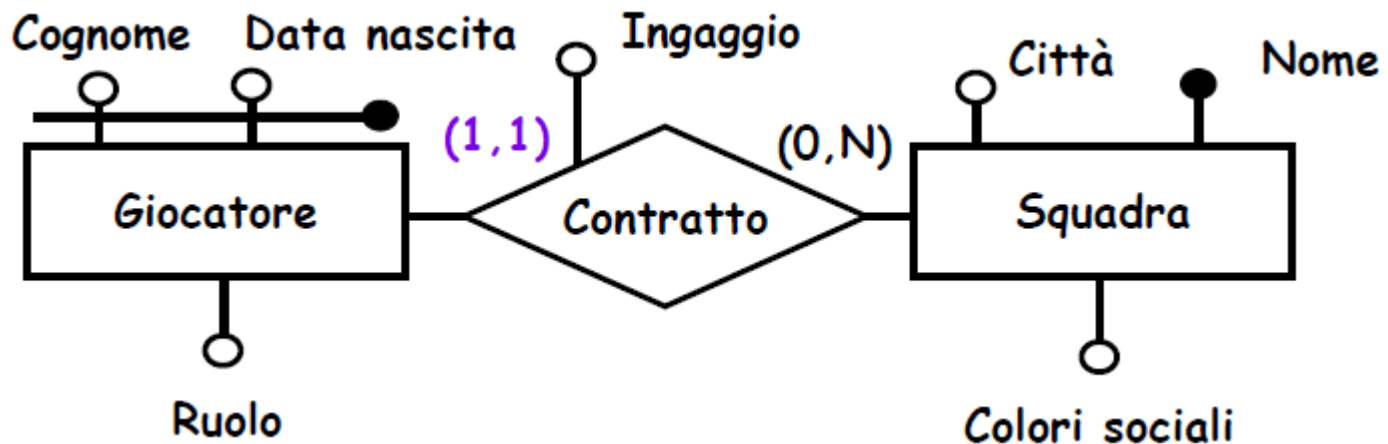
Prodotto

<u>Fornitore</u>	<u>Prodotto</u>	<u>Dipartimento</u>	Quantità
07092913	A0034	Fisica	5

<u>Nome</u>	Telefono
Ingegneria	0577233601
Fisica	0577232471
Biologia	0577234632

Dipartimento

# Associazioni uno a molti (1)



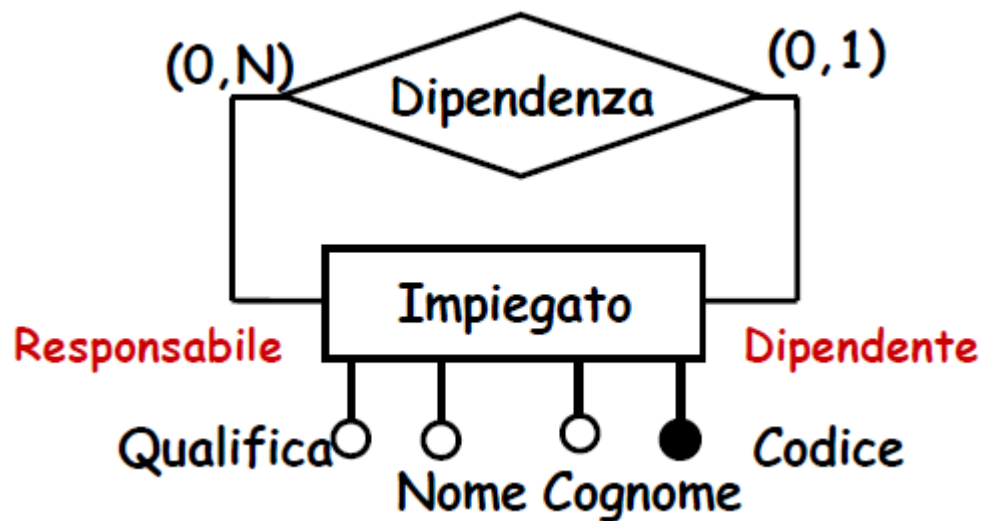
- ❑ **Giocatore** (Cognome, DataNascita, Ruolo)
- ❑ **Squadra** (Nome, Città, ColoriSociali)
- ❑ **Contratto** (CognGiocatore, DataNascG, Squadra, Ingaggio)
  - FK: (CognGiocatore, DataNascG) REFERENCES Giocatore
  - FK: Squadra REFERENCES Squadra
- ❑ Il Nome della Squadra non fa parte della chiave di Contratto

# Associazioni uno a molti (2)

- ❑ Poiché un giocatore ha un contratto con una sola squadra, nella relazione Contratto un giocatore non può apparire in più tuple.
- ❑ Si può pertanto adottare anche una **soluzione più compatta, che fa uso di 2 sole relazioni**:
- ❑ **Giocatore** (Cognome, DataNasc, Ruolo, Squadra, Ingaggio)
  - FK: Squadra REFERENCES Squadra
- ❑ **Squadra** (Nome, Città, ColoriSociali)
- ❑ che corrisponde a tradurre l'associazione insieme a Giocatore (ovvero all'entità che partecipa con cardinalità massima 1)
- ❑ Se fosse  $\text{min-card}(\text{Giocatore}, \text{Contratto}) = 0$ , allora gli attributi **Squadra** e **Ingaggio** dovrebbero entrambi ammettere valore nullo (**e per un giocatore o lo sono entrambi o non lo è nessuno dei due**).

# Associazioni ad anello uno a molti

- ❑ In questo caso è possibile operare una traduzione con 1 o 2 relazioni.

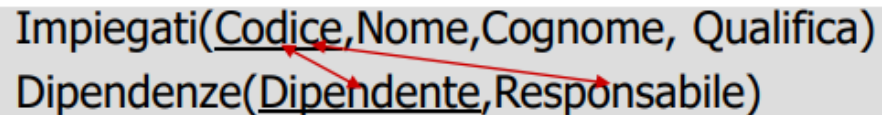


- ❑ **1 relazione:**
  - ❑ **Impiegato** (Codice, Nome, Cognome, Qualifica, Responsabile\*)
    - FK: Responsabile REFERENCES Impiegato
- ❑ **2 relazioni:**
  - ❑ **Impiegato** (Codice, Nome, Cognome, Qualifica)
  - ❑ **Dipendenza** (Dipendente, Responsabile)
    - FK: Dipendente REFERENCES Impiegato
    - FK: Responsabile REFERENCES Impiegato

# Associazioni ad anello uno a molti

2 relazioni:

Impiegati(Codice, Nome, Cognome, Qualifica)  
Dipendenze(Dipendente, Responsabile)

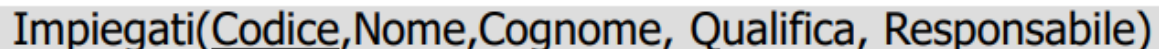


<u>Codice</u>	Nome	Cognome	Qualifica
D1	Mario	Bianchi	Dirigente
D2	Luca	Rossi	Capo-Reparto
D3	Gianni	Neri	Operaio
D4	Davide	Verdi	Operaio

<u>Dipendente</u>	Responsabile
D2	D1
D3	D2
D4	D2

1 relazione:

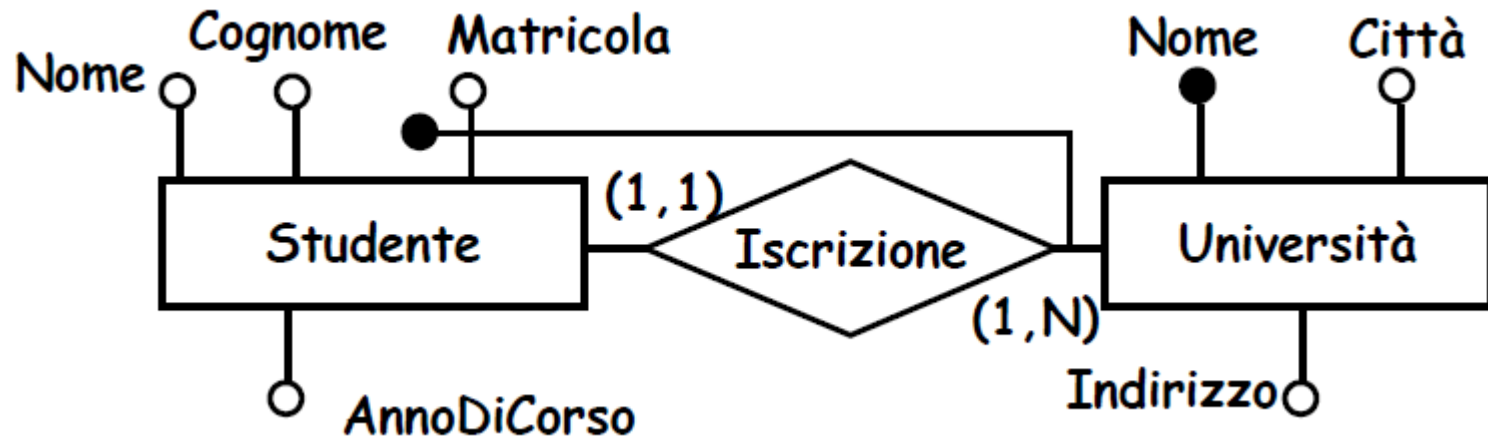
Impiegati(Codice, Nome, Cognome, Qualifica, Responsabile)



<u>Codice</u>	Nome	Cognome	Qualifica	Responsabile
D1	Mario	Bianchi	Dirigente	null
D2	Luca	Rossi	Capo-Reparto	D1
D3	Gianni	Neri	Operaio	D2
D4	Davide	Verdi	Operaio	D2

# Entità con identificazione esterna

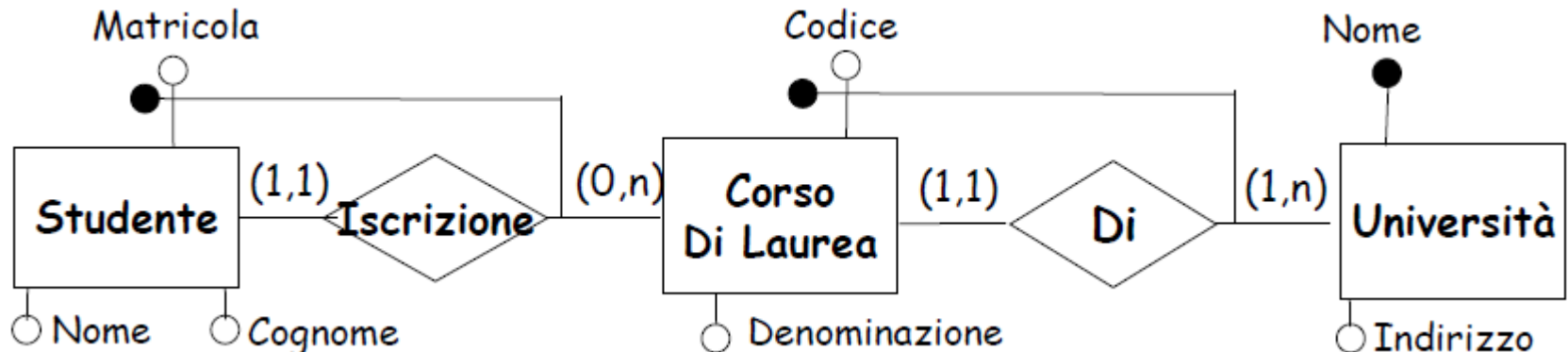
- ❑ Nel caso di entità identificata esternamente, si “importa” l’identificatore della/e entità identificante/i.
- ❑ L’associazione relativa risulta automaticamente tradotta.



- ❑ **Studente** (Matricola, Università, Cognome, Nome, AnnoDiCorso)  
➤ FK: Università REFERENCES Università
- ❑ **Università** (Nome, Città, Indirizzo)

# Identificazioni esterne: una precisazione

- ❑ Nel caso generale, si possono avere **identificazioni esterne in cascata**.
- ❑ Per operare correttamente occorre **partire dalle entità non identificate esternamente** e propagare gli identificatori che così si ottengono.

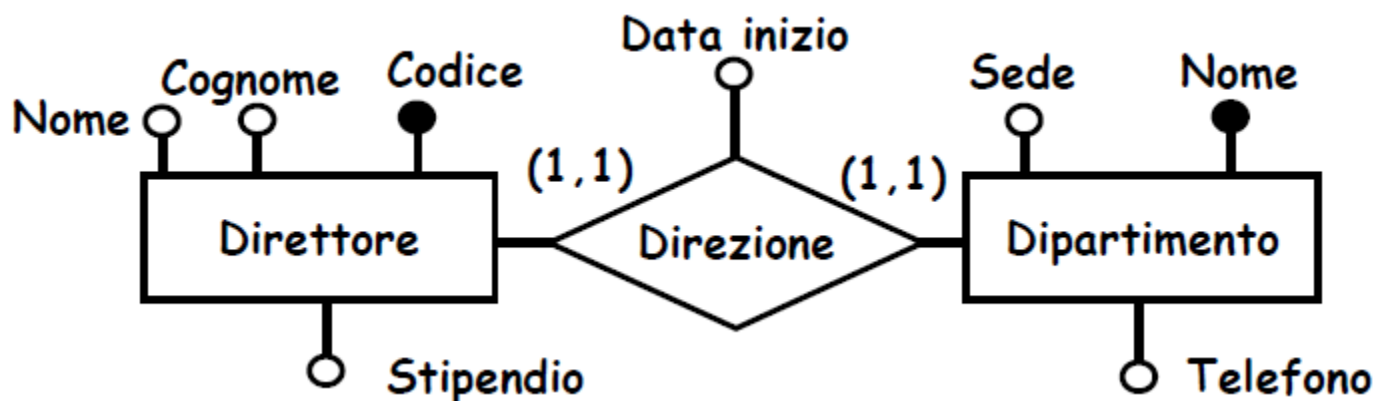


- ❑ **Università** (Nome, Indirizzo)
- ❑ **CorsoDiLaurea** (Università, Codice, Denominazione)
- ❑ **Studente** (Università, CodiceCdL, Matricola, Cognome, Nome)



# Associazioni uno a uno (1)

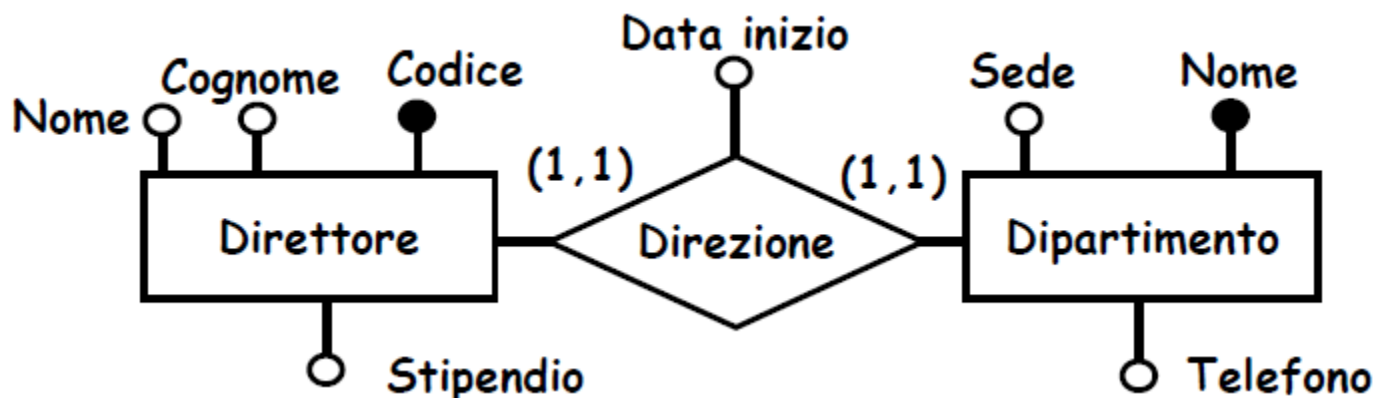
- ❑ Si hanno a disposizione varie possibilità (traduzione con 1, 2 o 3 relazioni)



- ❑ **Tre relazioni:**
- ❑ **Direttore** (Codice, Nome, Cognome, Stipendio)
- ❑ **Dipartimento** (Nome, Sede, Telefono)
- ❑ **Direzione** (Direttore, Dipartimento, DataInizio)
  - FK:...
  - Unique(Dipartimento)

L'identificatore di una delle due entità è scelto come chiave primaria, l'altro dà origine a una chiave alternativa. La scelta dipende dall'importanza relativa delle chiavi.

# Associazioni uno a uno (2)



## ❑ Due relazioni:

❑ **Direttore** (Codice, Nome, Cognome, Stipendio, Dipartimento, DataInizio)

➤ FK: Dipartimento REFERENCES Dipartimento

➤ Unique(Dipartimento)

❑ **Dipartimento** (Nome, Sede, Telefono)

oppure

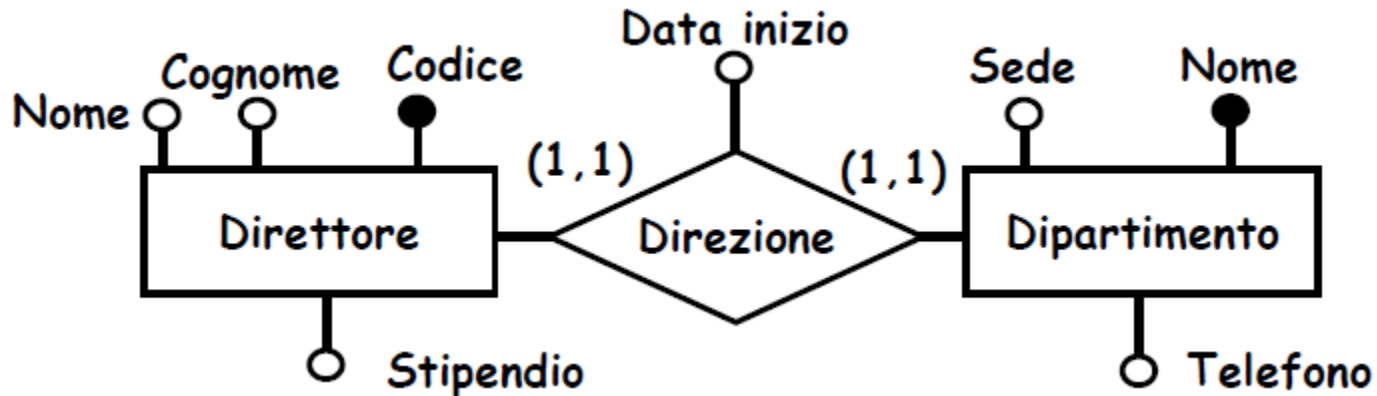
❑ **Direttore** (Codice, Nome, Cognome, Stipendio)

❑ **Dipartimento** (Nome, Sede, Telefono, Direttore, DataInizio)

➤ FK: Direttore REFERENCES Direttore

➤ Unique(Direttore)

# Associazioni uno a uno (3)



## □ Una relazione:

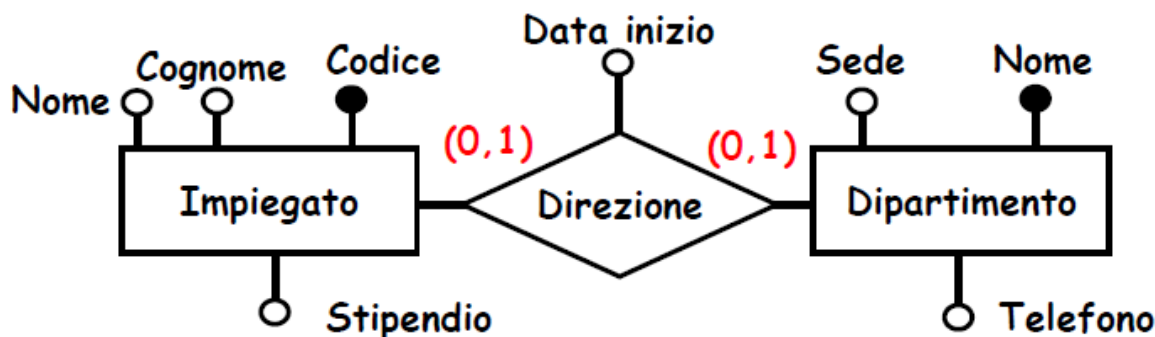
- **Direttore** (Codice, Nome, Cognome, Stipendio, DataInizio, Dipartimento, Sede, Telefono)
  - Unique(Dipartimento)

oppure

- **Dipartimento** (Nome, Sede, Telefono, Direttore, NomeDir, CognomeDir, Stipendio, DataInizio)
  - Unique(Direttore)

# Associazioni uno a uno con opzionalità

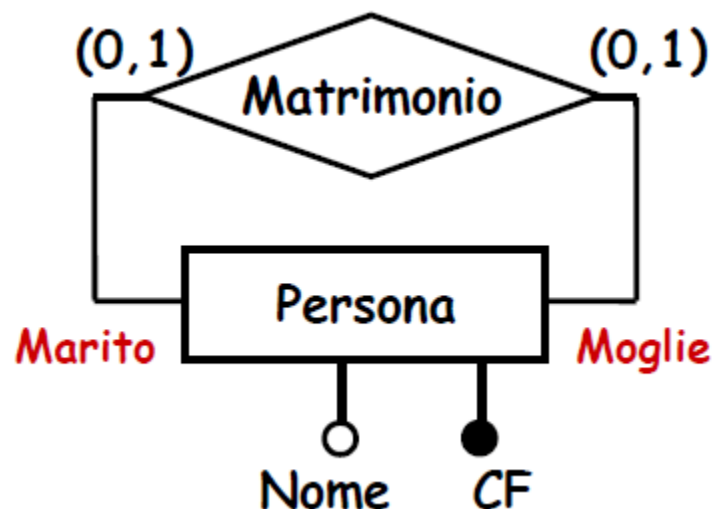
- ❑ La traduzione con una sola relazione corrisponde a un accorpamento di entità:
  - Se  $\text{min-card}(E1,R) = \text{min-card}(E2,R) = 1$  si avranno due chiavi, entrambe senza valori nulli (la chiave primaria è “la più importante”);
  - Se  $\text{min-card}(E1,R) = 0$  e  $\text{min-card}(E2,R) = 1$  la chiave derivante da E2 ammetterà valori nulli, e la chiave primaria si ottiene da E1;
  - Se  $\text{min-card}(E1,R) = \text{min-card}(E2,R) = 0$  entrambe le chiavi hanno valori nulli, quindi si rende necessario introdurre un codice.



- ❑ **ImpDip** (CodiceImpDip, CodiceImp\*, ..., Dipartimento\*, ..., DataInizio\*)

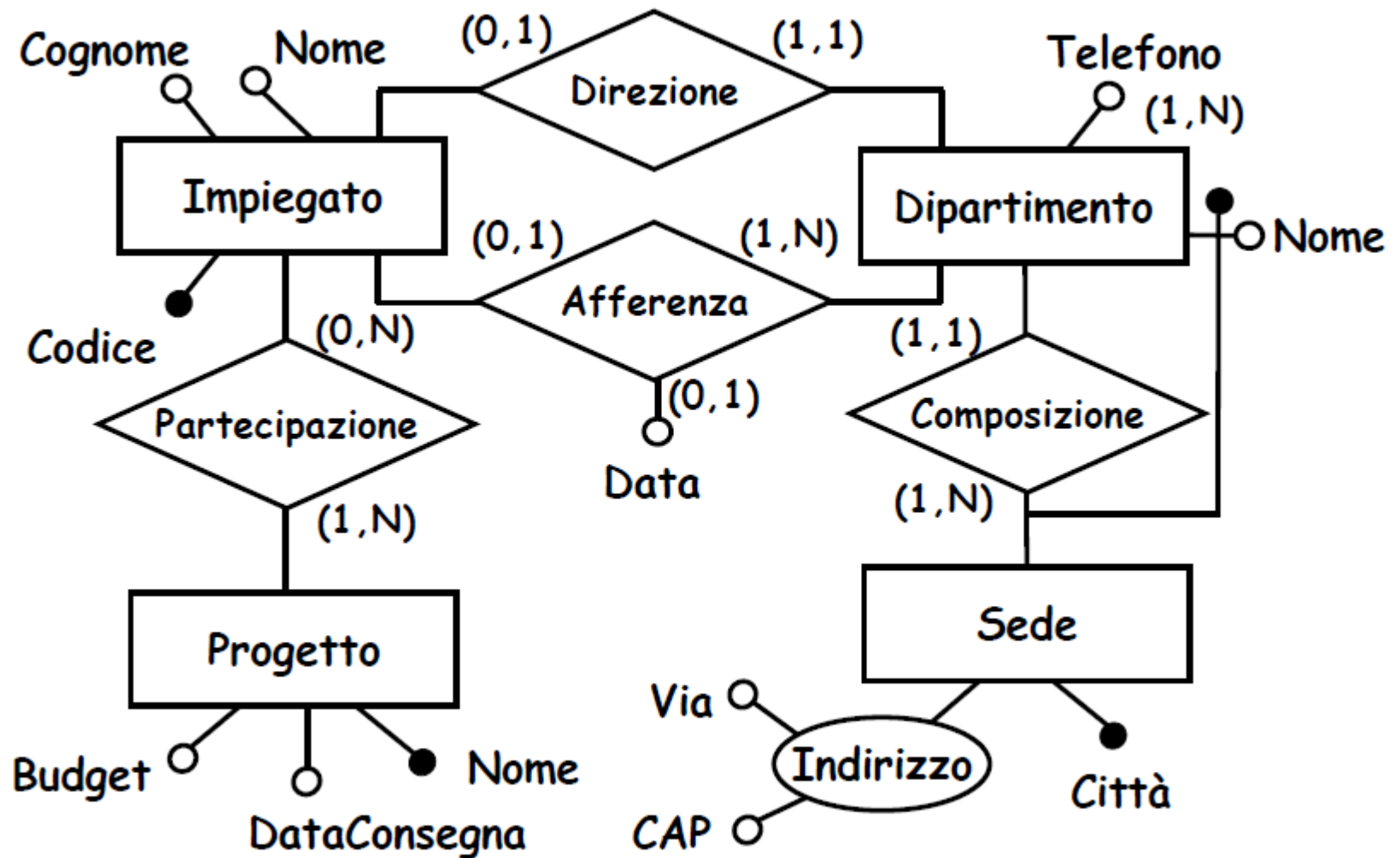
# Associazioni ad anello uno a uno

- ❑ In questo caso è possibile operare una traduzione con una o due relazioni
- ❑ La traduzione con una relazione è ancora problematica se entrambe le partecipazioni sono opzionali

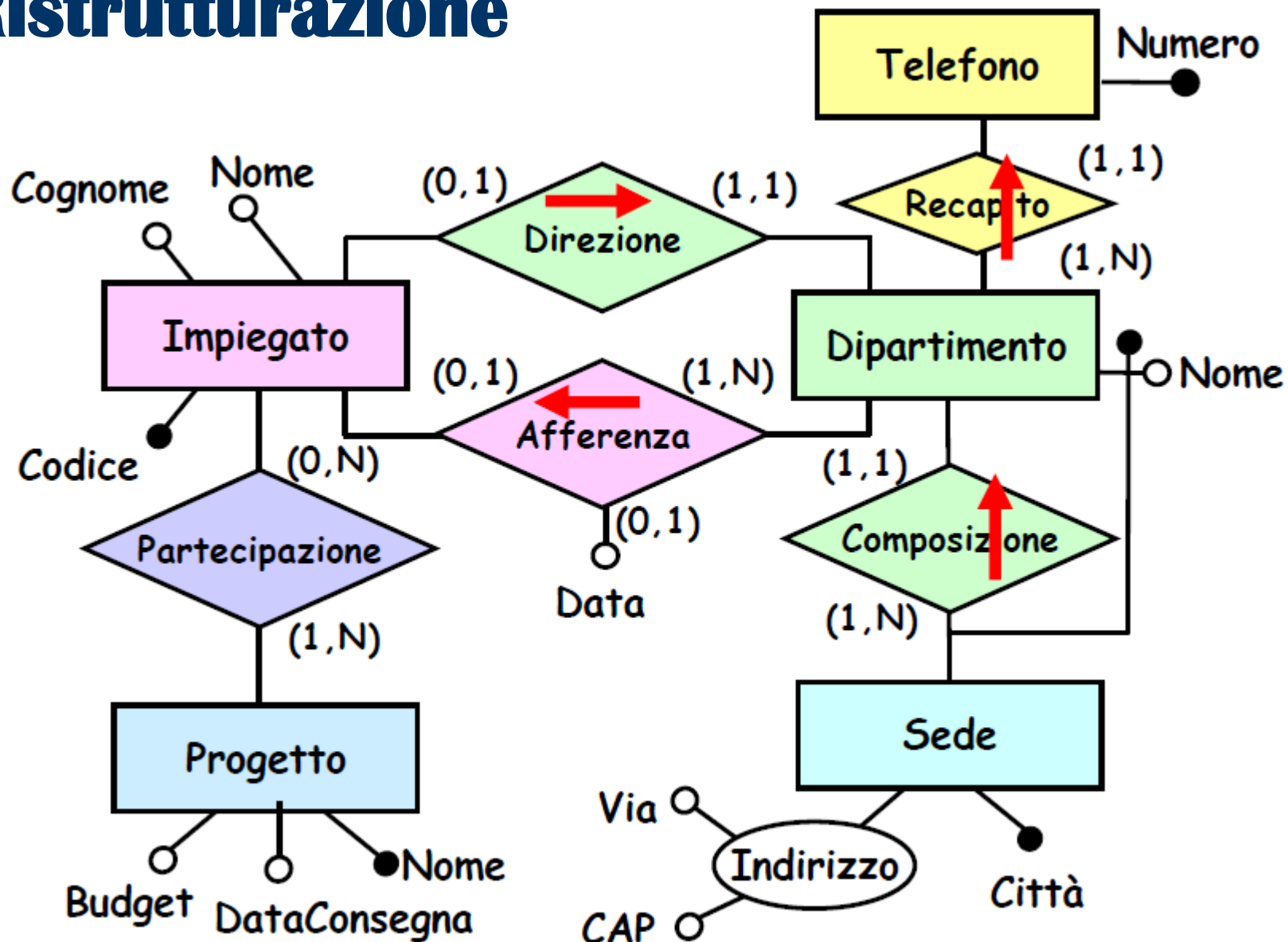


- ❑ **Una relazione:**
- ❑ **Persona** (Codice, CFUomo\*, NomeUomo\*, CFDonna\*, NomeDonna\*)
- ❑ **Due relazioni:**
- ❑ **Persona** (CF, Nome)
- ❑ **Matrimonio** (Marito, Moglie)
  - FK: Marito REFERENCES Persona
  - FK: Moglie REFERENCES Persona Unique (Moglie)

# Esempio di riferimento



# Ristrutturazione



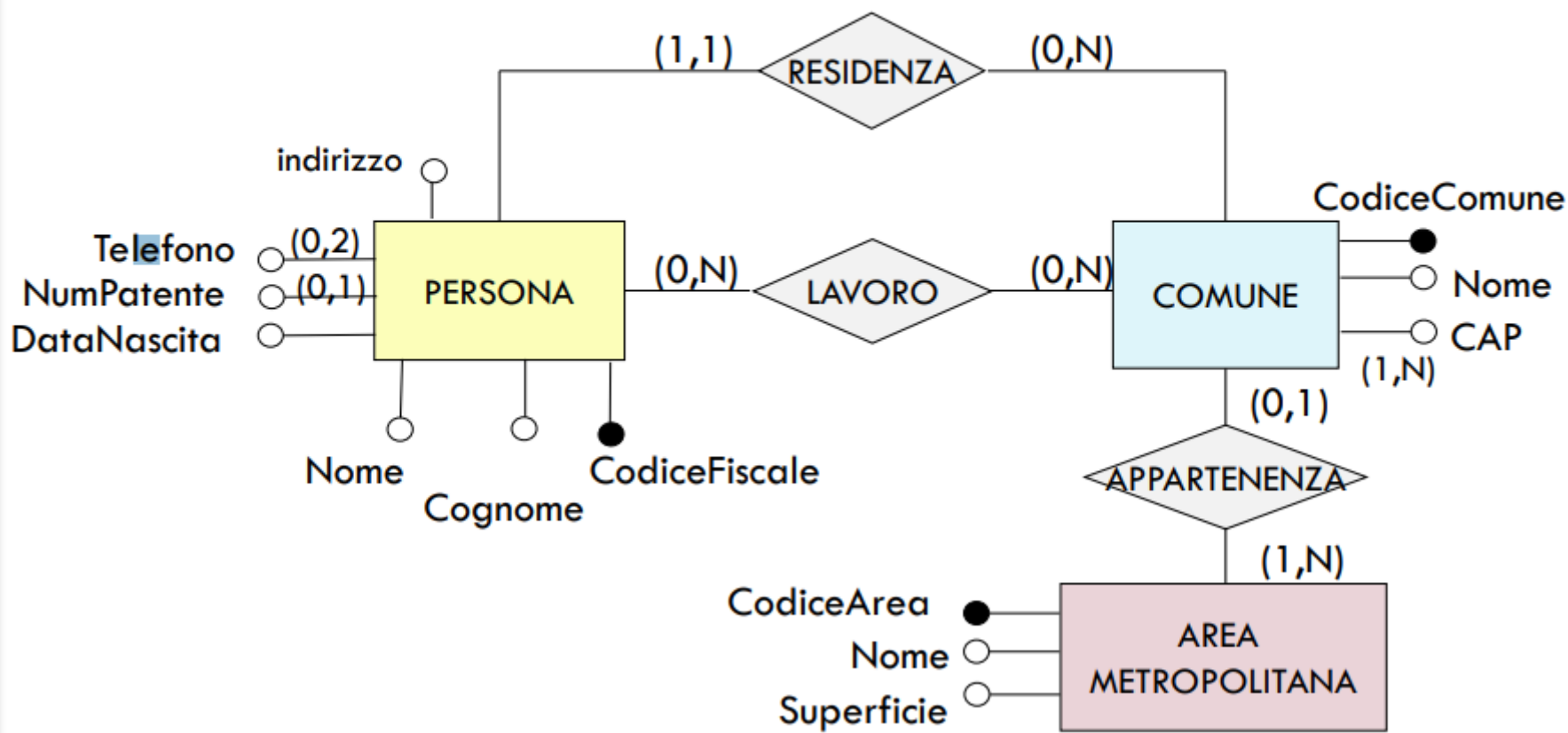
# Schema logico relazionale

- ❑ Per le entità E che partecipano ad associazioni sempre con  $\max\text{-card}(E,R) = n$  la traduzione è immediata:
  - **Sede** (Città, Via, CAP)
  - **Progetto** (Nome, Budget, DataConsegna)
- ❑ Anche l'associazione Partecipazione si traduce immediatamente:
  - **Partecipazione** (Impiegato, Progetto)
  - FK: Impiegato REFERENCES Impiegato
  - FK: Progetto REFERENCES Progetto
- ❑ L'entità Dipartimento si traduce importando l'identificatore di Sede e inglobando l'associazione Direzione:
  - **Dipartimento** (Nome, Città, Direttore)
  - FK: Città REFERENCES Sede
  - FK: Direttore REFERENCES Impiegato
- ❑ L'entità Telefono si traduce con una relazione che ingloba l'associazione Recapito
  - **Telefono** (Numero, Nome, Città)
  - FK: Nome, Città REFERENCES Dipartimento
- ❑ Per tradurre l'associazione Afferenza, assumendo che siano pochi gli impiegati che non afferiscono a nessun dipartimento, si opta per una rappresentazione compatta
  - **Impiegato** (Codice, Nome, Cognome, NomeDip\*, CittàDip\*, Data\*)
  - FK: NomeDip, CittàDip REFERENCES Dipartimento

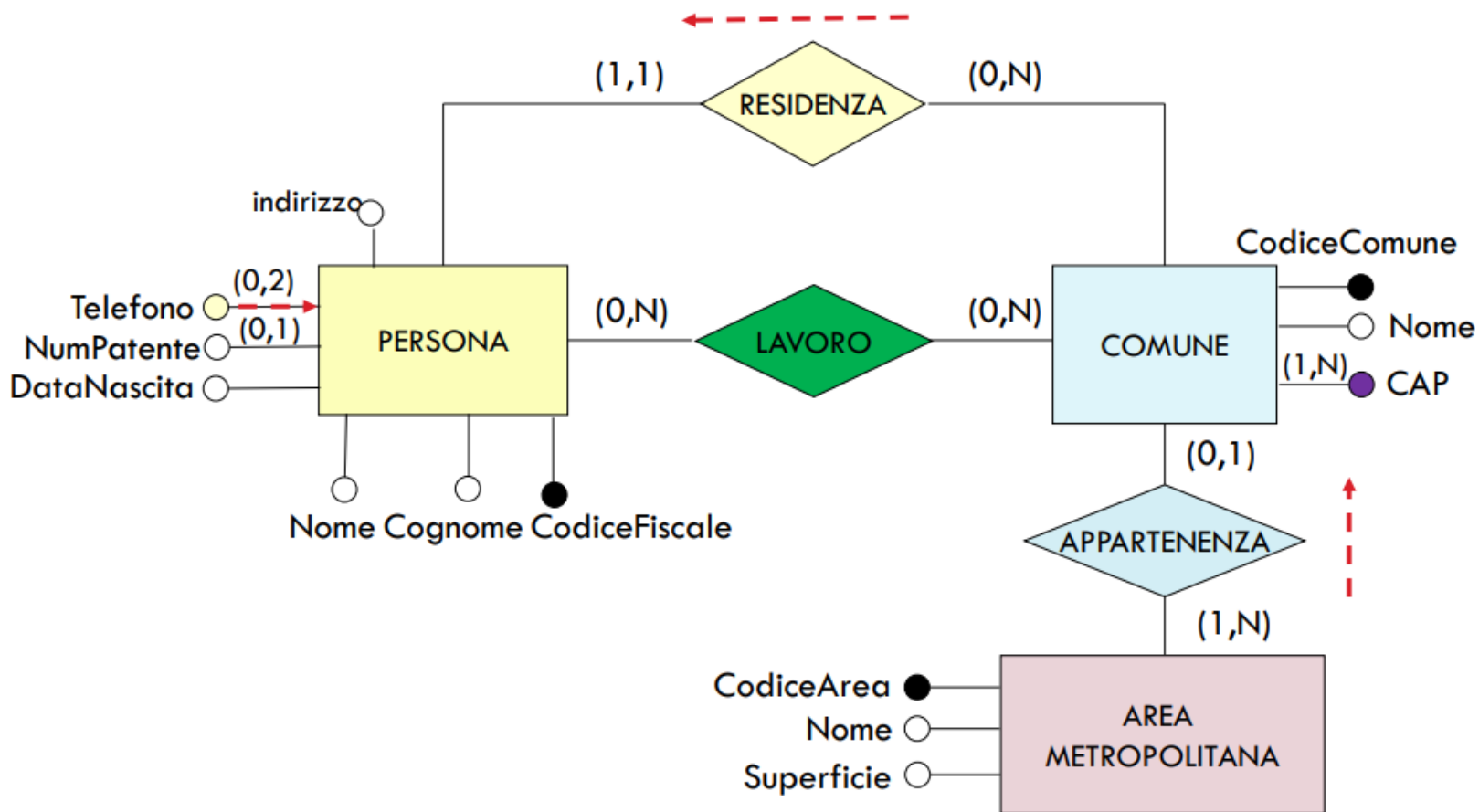


# Esercizio 4






Tradurre lo schema nel modello relazionale



# Esercizio 4 - soluzione



# Esercizio 4 - soluzione

-  PERSONA(CE, Nome, Cognome, Patente, DataNascita, Tel1, Tel2, indirizzo, CodiceComune:COMUNE)
-  COMUNE(CodiceCatastale, Nome, CodiceArea\*:AREAMETROPOLITANA)
-  CAP(CAP, CodComune:COMUNE)
-  LAVORO(CE:PERSONA, CodComune:COMUNE)
-  AREAMETROPOLITANA(CadiceArea, Nome, Superficie)

# Osservazioni finali

- ❑ La progettazione logica non deve essere condotta “alla cieca”; in presenza di diverse alternative occorre valutare diversi fattori, tra cui:
  - la presenza o meno di valori nulli, e la loro incidenza, che dipende dal **volume dei dati**;
  - le porzioni di schema E/R interessate dalle varie **operazioni** (con particolare riferimento ai join tra le relazioni che vengono create);
  - la flessibilità degli schemi relazionali rispetto ad evoluzioni future.
- ❑ I casi visti (**semplici esempi a scopo didattico**) non esauriscono certamente l'argomento e lasciano sempre spazio per soluzioni specifiche “ad hoc”.
- ❑ Ad esempio, associazioni uno a molti con  $\text{max-card}(E2,R) = K$ , con  $K$  “piccolo”, possono al limite essere tradotte con 1 sola relazione, prevedendo  $K$  repliche degli attributi di  $E2$  (es. tipico: numeri di telefono).

# Sommario

- ❑ La **fase di progettazione logica** ha lo scopo di derivare uno schema logico che rispetti quanto più possibile i concetti espressi nello schema E/R di partenza e che sia al tempo spesso “efficiente”.
- ❑ I confronti tra le diverse alternative sono eseguiti considerando le **principali operazioni** interessate e i **volumi dei dati** in gioco.
- ❑ La fase di **ristrutturazione** elimina dallo schema E/R tutti i costrutti che non possono essere direttamente rappresentati nel modello logico, e apporta modifiche strutturali sulla base di considerazioni di efficienza.
- ❑ La fase di **traduzione** opera traducendo entità e associazioni.
- ❑ Le diverse alternative che si hanno a disposizione per tradurre le associazioni dipendono dalle **cardinalità massime** in gioco, le quali determinano anche le chiavi delle relazioni che si ottengono.
- ❑ Le **cardinalità minime** possono portare, in funzione della traduzione scelta, ad avere valori nulli.