

Laboratorio di Basi di Dati

Progettazione Logica

Luca Anselma

anselma@di.unito.it

Outline


- Progettazione logica
- Ristrutturazione dello schema concettuale
- Traduzione nel modello logico

Progettazione logica

- È la fase successiva alla progettazione concettuale
- *Obiettivo*: “tradurre” lo schema concettuale in uno schema logico che rappresenti gli stessi dati in maniera corretta ed efficiente
- Non si tratta di una traduzione semplice e immediata:
 1. Alcuni aspetti non sono direttamente rappresentabili
 2. È necessario considerare le prestazioni

Progettazione logica

Dati di ingresso e di uscita

- Ingresso:
 - schema concettuale
 - informazioni sul carico applicativo
 - modello logico che si intende usare

- Uscita:
 - schema logico
 - vincoli di integrità
 - documentazione associata

Progettazione logica

È suddivisibile in due fasi:

- **Ristrutturazione dello schema concettuale (EER):**
fase in cui viene nuovamente analizzato lo schema EER al fine di evidenziare ed eliminare alcune "inefficienze". Le inefficienze sono normali visto che nella progettazione concettuale ci si concentra sulla realtà da modellare e non su ottimizzazioni
- **Traduzione verso il modello logico e ottimizzazioni:**
fase in cui lo schema EER è tradotto in relazionale (tramite un algoritmo). In fase di traduzione è possibile applicare delle ottimizzazioni (per es. la normalizzazione) che producono schemi leggermente differenti e maggiormente adatti alle condizioni ipotizzate

Progettazione logica

Schema E-R principale
e regole
aziendali



Ristrutturazione
dello schema E-R



Carico
applicativo



Schema E-R
ristrutturato e
regole aziendali



Modello
logico



Traduzione nel
modello logico



Schema
logico

Ristrutturazione dello schema concettuale

Ristrutturazione schema E-R

- Obiettivi:
 - semplificare la traduzione nel modello logico
 - “ottimizzare” le prestazioni
- Osservazione:
 - uno schema E-R ristrutturato non è (più) uno schema concettuale nel senso stretto del termine perché tiene conto degli aspetti realizzativi

Premessa: analisi delle prestazioni

Per ottimizzare il risultato abbiamo bisogno di analizzare le prestazioni a questo livello

Ma:

le prestazioni non sono valutabili con precisione su uno schema concettuale perché dipendono anche da parametri fisici!

Quindi usiamo degli indicatori di massima, che permettono di valutare il carico applicativo

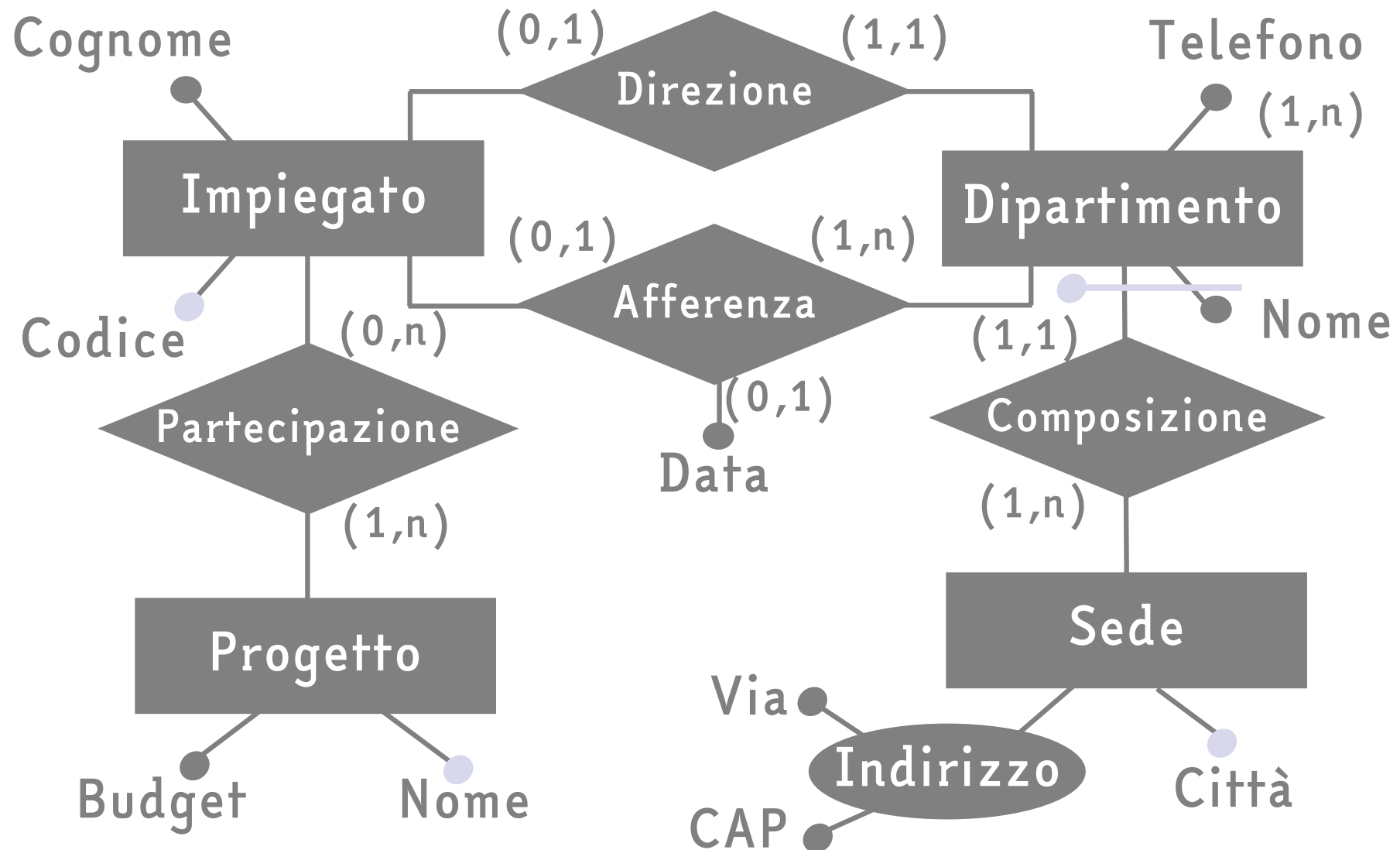
“Indicatori” di prestazione

- **tempo:** numero di occorrenze di entità e di associazioni visitate per eseguire un'operazione sul DB
- **spazio:** spazio di memoria necessario per rappresentare i dati

Per poterli valutare abbiamo bisogno di alcune informazioni:

- **volume dei dati:** numero di occorrenze, dimensione degli attributi
- **caratteristiche delle operazioni:** operazione interattiva/batch, frequenza, entità/associazioni coinvolte

Esempio



Esempio di tavola dei volumi



Concetto	Tipo	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



Esempio di tavola delle operazioni



Operazione	Descrizione	Tipo	Frequenza
1	Assegna un impiegato a un progetto	I	50 al giorno
2	Trova tutti i dati di un impiegato, del dipartimento in cui lavora e dei progetti ai quali partecipa	I	100 al giorno
3	Trova i dati di tutti gli impiegati di un certo dipartimento	I	10 al giorno
4	Per ogni sede, trova i suoi dipartimenti con il cognome del direttore e l'elenco degli impiegati	B	2 a settimana

Di solito è sufficiente considerare solo le operazioni principali per il principio di Pareto "80/20": l'80% del carico viene generato dal 20% delle operazioni

Ristrutturazione dello schema concettuale: passi da seguire

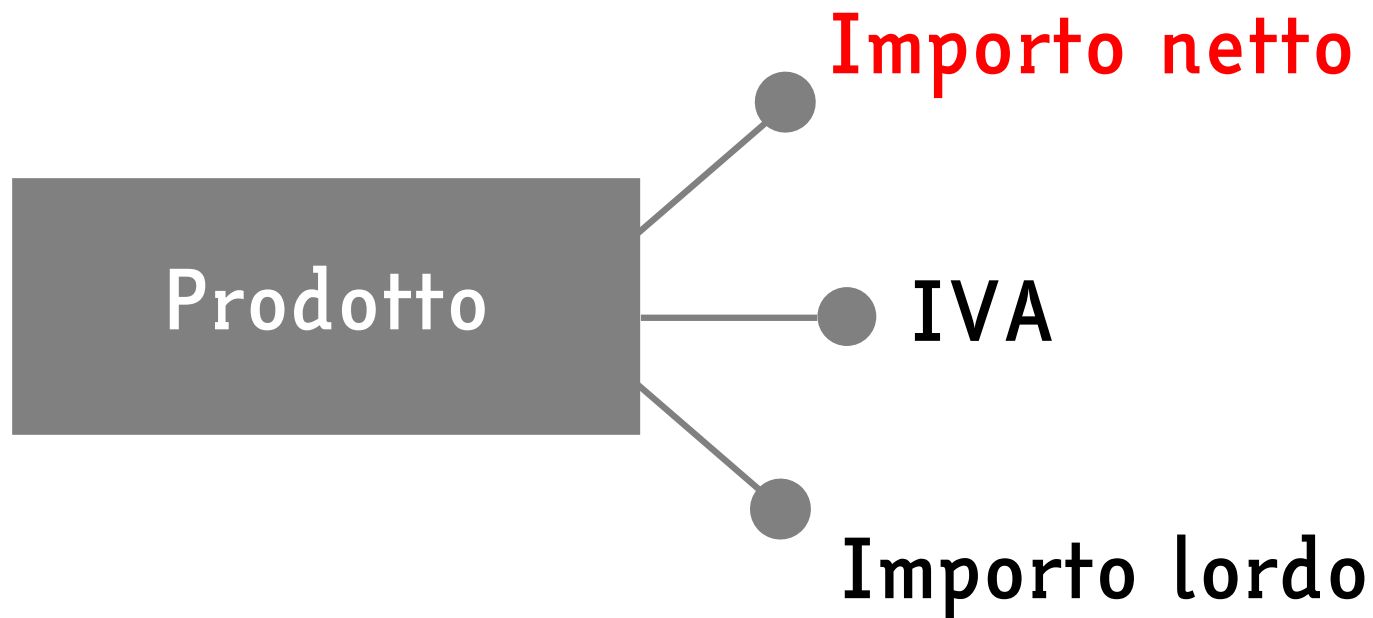
1. **Analisi delle ridondanze:** si decide se eliminare o aggiungere ridondanze presenti nello schema
2. **Eliminazione delle generalizzazioni:** tutte le generalizzazioni presenti vengono analizzate e sostituite da altri concetti
3. **Partizionamento/accorpamento di entità e associazioni:** si decide se è opportuno partizionare o accorpare concetti dello schema in unico concetto
4. **Scelta degli identificatori principali:** si sceglie un identificatore per le entità che ne hanno più di uno
5. **Eliminazione degli attributi multivalore**
6. **Eliminazione degli attributi composti**

1. 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

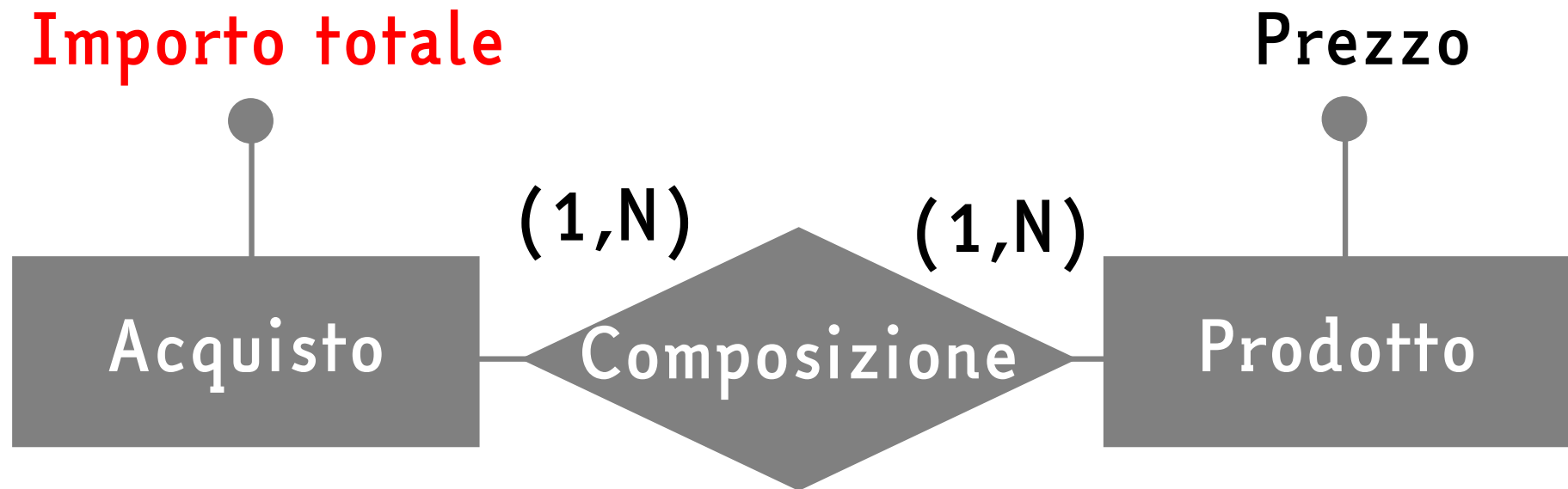
Forme di ridondanza in uno schema E-R

- Attributi derivabili:
 - da altri attributi della stessa entità o associazione



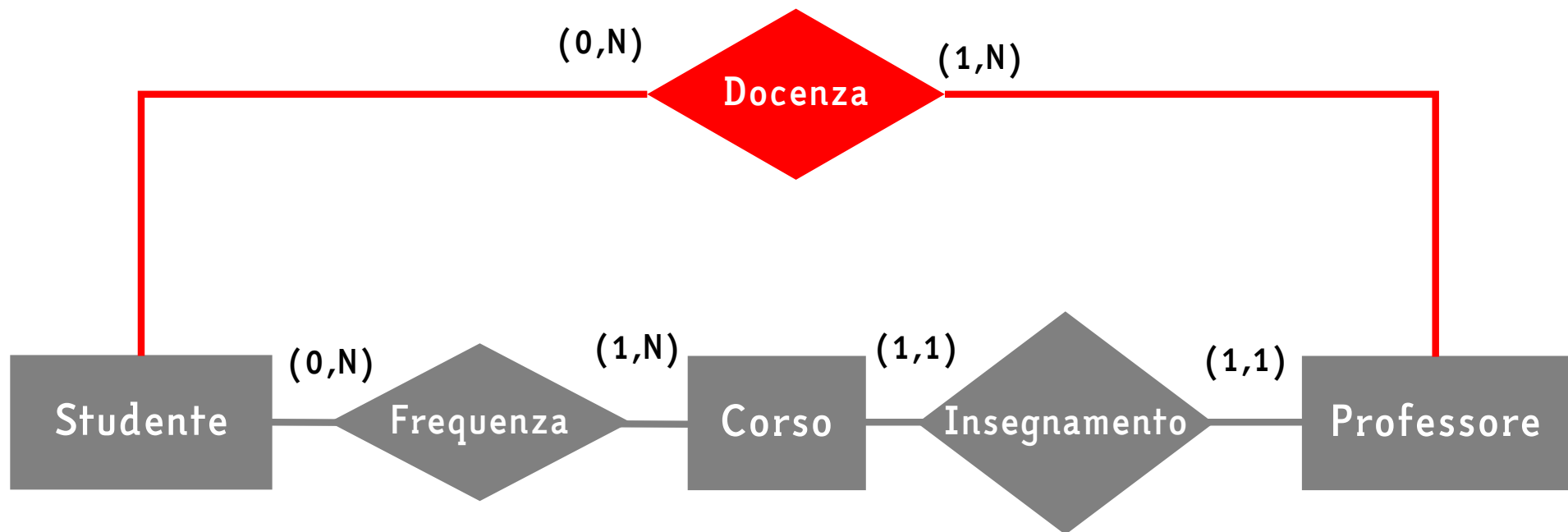
Forme di ridondanza in uno schema E-R

- Attributi derivabili:
 - da attributi di altre entità o associazioni



Forme di ridondanza in uno schema E-R

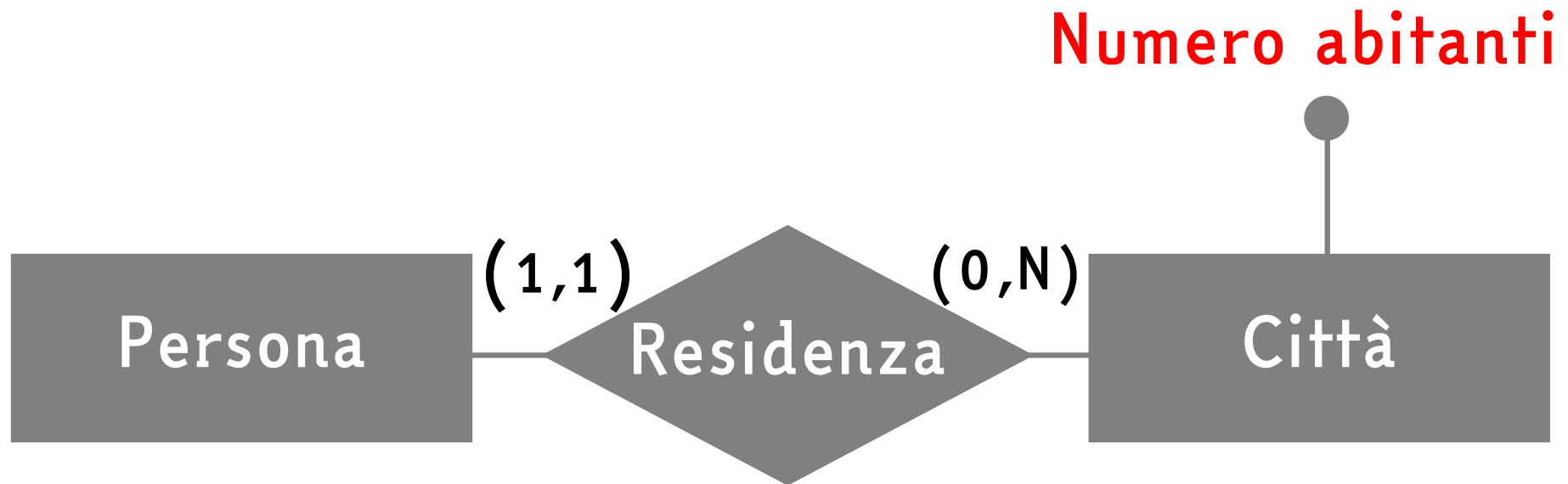
- Associazioni derivabili dalla composizione di altre associazioni in presenza di cicli



Vantaggi e svantaggi

- Operazioni di interrogazione/lettura dei dati
 - Più efficienti
 - Semplificate
- Operazioni di inserimento e modifica dei dati
 - Meno efficienti
- Maggiore occupazione di spazio

Esempio di analisi di una ridondanza (1)



Analizziamo le prestazioni con la presenza e l'assenza della ridondanza e confrontiamo i risultati

Esempio di analisi di una ridondanza (2)

Tavola dei volumi

Concetto	Tipo	Volume
Città	E	200
Persona	E	1 000 000
Residenza	A	1 000 000

Tavola delle operazioni

Operazione	Descrizione	Tipo	Frequenza
1	Memorizza una nuova persona con la relativa città di residenza	I	500 al giorno
2	Stampa tutti i dati di una città (incluso il numero di abitanti)	B	2 al giorno

Scenario A: Assenza di ridondanza

Operazione 1

Schema di operazione

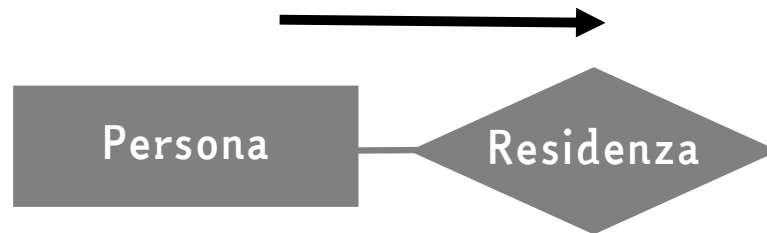


Tavola degli accessi

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



Scenario A: Assenza di ridondanza

Operazione 2

Schema di operazione

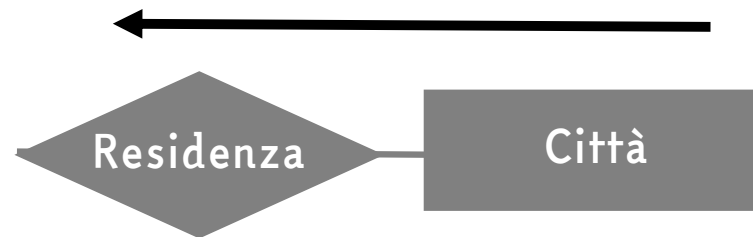


Tavola degli accessi

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



Scenario B: Presenza di ridondanza

Operazione 1

Schema di operazione



Tavola degli accessi

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



Scenario B: Presenza di ridondanza

Operazione 2

Schema di operazione

Numero abitanti



Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
Città	Entità	1	L

Scenario A: Assenza di ridondanza

Costi:

- Spazio: 0 byte
- Tempo:
 - Operazione 1: $2 * 500$ accessi in scrittura al giorno
 - Operazione 2: $5000 * 2$ accessi in lettura al giorno
 - Contando doppi gli accessi in scrittura:
Totale di 12000 accessi al giorno

Scenario B: Presenza di ridondanza

Costi:

- **Spazio:** assumendo di usare 4 byte per memorizzare il numero di abitanti di ogni città ($\sim 2^{32}$ abitanti): $4 * 200$ byte
- **Tempo:**
 - Operazione 1: 3 * 500 accessi in scrittura e 500 accessi in lettura al giorno
 - Operazione 2: trascurabile
 - Assumendo che un accesso in scrittura abbia un costo doppio rispetto a un accesso in lettura:
Totale di 3500 accessi al giorno

Esempio di analisi di una ridondanza

- Eliminando la ridondanza risparmieremmo meno di un kilobyte ma avremmo circa 8500 accessi al giorno in più al database
- Quindi possiamo concludere che, in questo esempio, mantenere aggiornato il dato ridondante costa molto meno che ricalcolarlo

Analisi delle ridondanze

- Per ogni ridondanza:
 - Per ogni operazione significativa su cui la presenza/assenza della ridondanza può avere effetto:
 - Schema di operazione in presenza e in assenza di ridondanza
 - Tavola degli accessi in presenza e in assenza di ridondanza
 - Confronto in spazio e tempo tra presenza e assenza di ridondanza
 - Scelta se introdurre o non introdurre la ridondanza con motivazione

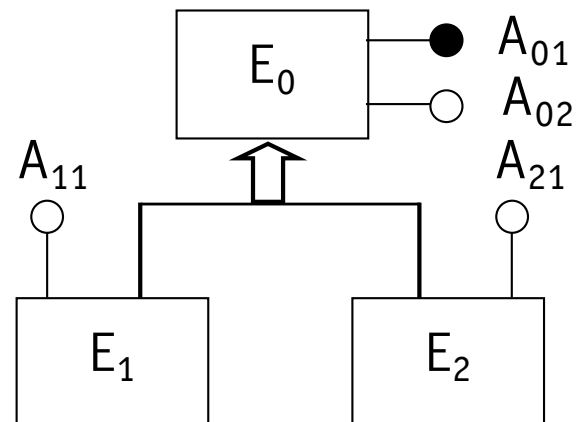
2. Eliminazione delle generalizzazioni

- Entità e associazioni sono direttamente rappresentabili nel modello relazionale
- Le generalizzazioni, invece, non sono direttamente rappresentabili

Si eliminano perciò le generalizzazioni sostituendole con entità, associazioni e regole aziendali

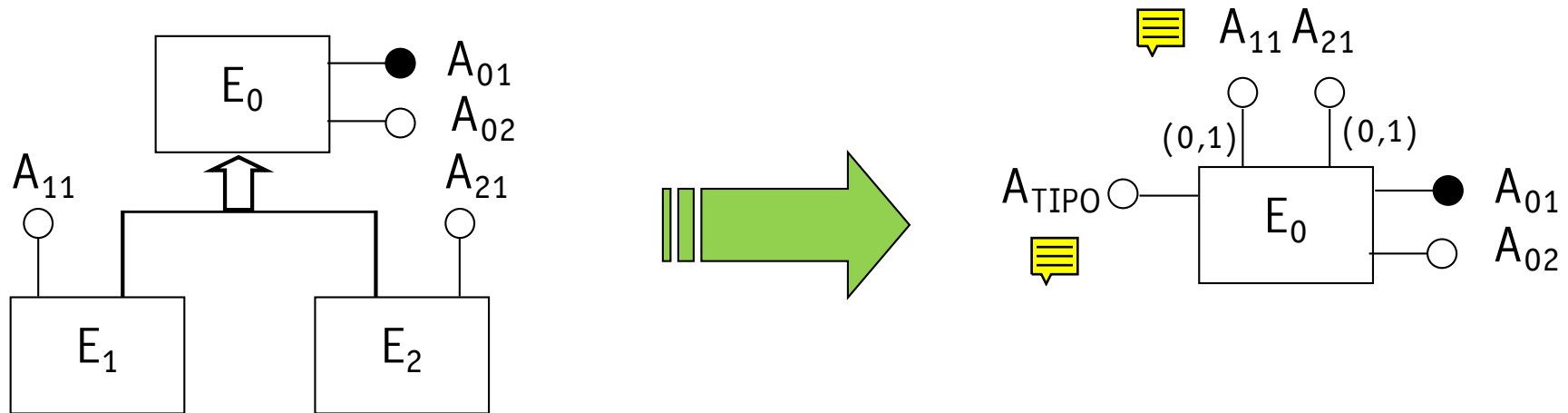
Tre possibilità

- ☞ i. Accorpamento dei figli della generalizzazione nel genitore
- ☞ ii. Accorpamento del genitore della generalizzazione nei figli
- ☞ iii. Sostituzione della generalizzazione con associazioni



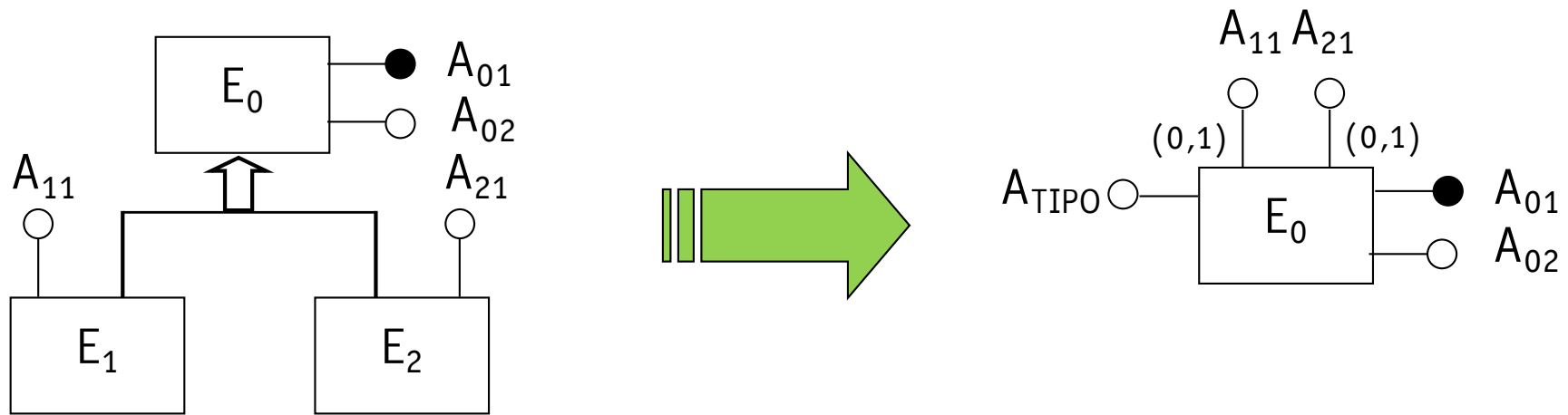
i) Accorpamento delle entità figlie nel genitore

Come ristrutturare:



- L'attributo A_{TIPO} serve a distinguere le istanze delle due entità figlie
- Se la generalizzazione è totale e/o esclusiva, bisogna aggiungere regole aziendali
- Se la generalizzazione è esclusiva, occorre aggiungere il vincolo che A_{11} e A_{21} non possono essere contemporaneamente *valorizzati*
- Se la generalizzazione è totale, occorre aggiungere il vincolo che A_{11} e A_{21} non possono essere contemporaneamente *non valorizzati*

i) Accorpamento delle entità figlie nel genitore



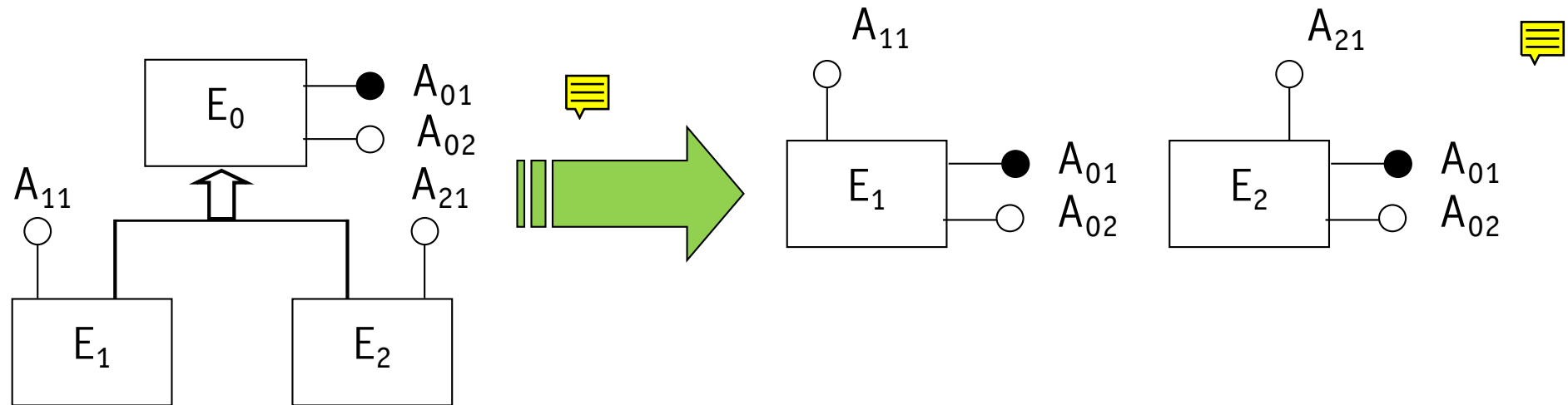
Considerazioni:

- Questa alternativa è conveniente quando le operazioni non fanno troppa distinzione tra le istanze delle varie entità. Le transazioni accederebbero infatti a un'unica tabella relazionale
- Spreca spazio producendo una base di dati con valori nulli



ii) Accorpamento del genitore nelle entità figlie

Come ristrutturare:



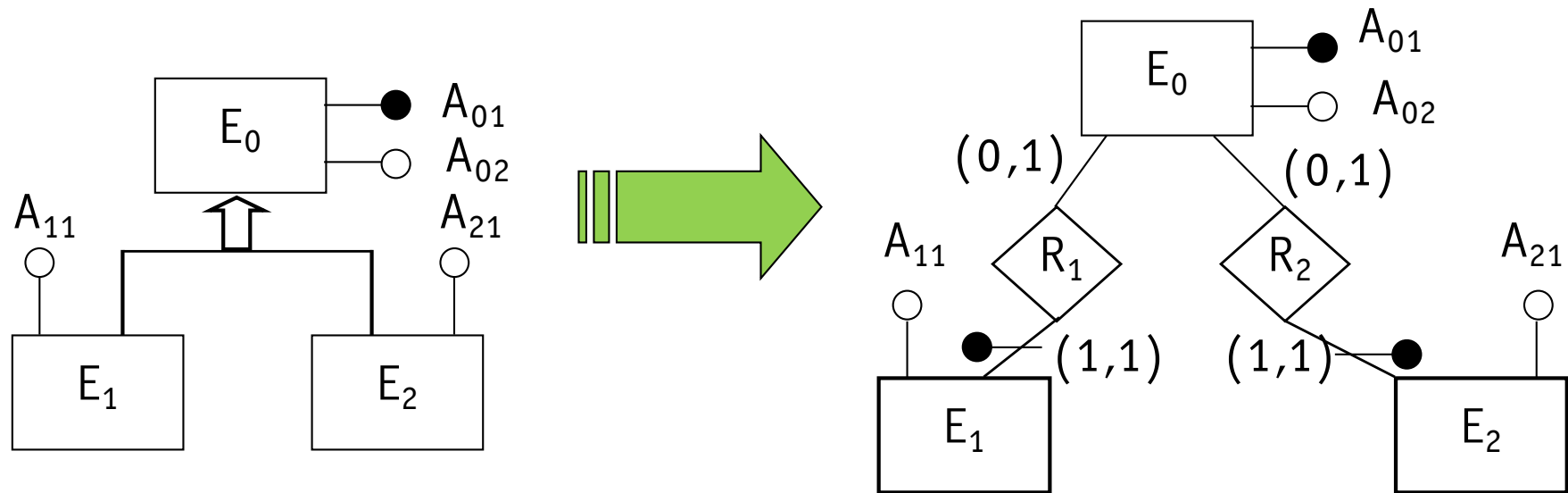
- È possibile soltanto se la generalizzazione è a partecipazione totale

Considerazioni:

- È conveniente quando le operazioni effettuano pochi accessi alle istanze di entrambe le entità ovvero le operazioni fanno ampiamente distinzione tra le entità figlie
- Ottimizza maggiormente lo spazio rispetto alle altre alternative perché usa meno attributi e associazioni

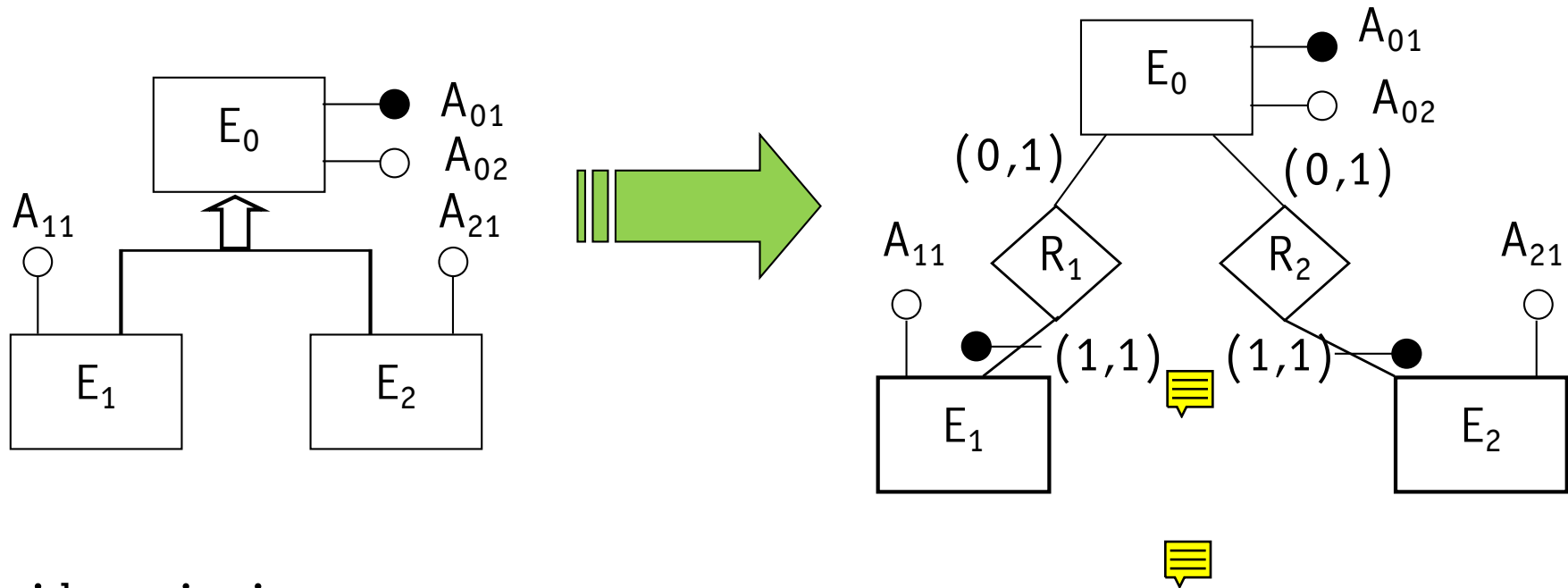
iii) Sostituzione della generalizzazione con associazioni

Come ristrutturare:



- Se la generalizzazione è esclusiva, occorre aggiungere il vincolo che ogni occorrenza di E_0 non può partecipare contemporaneamente a R_1 e a R_2
- Se la generalizzazione è totale, occorre aggiungere il vincolo che ogni occorrenza di E_0 deve partecipare a R_1 o a R_2

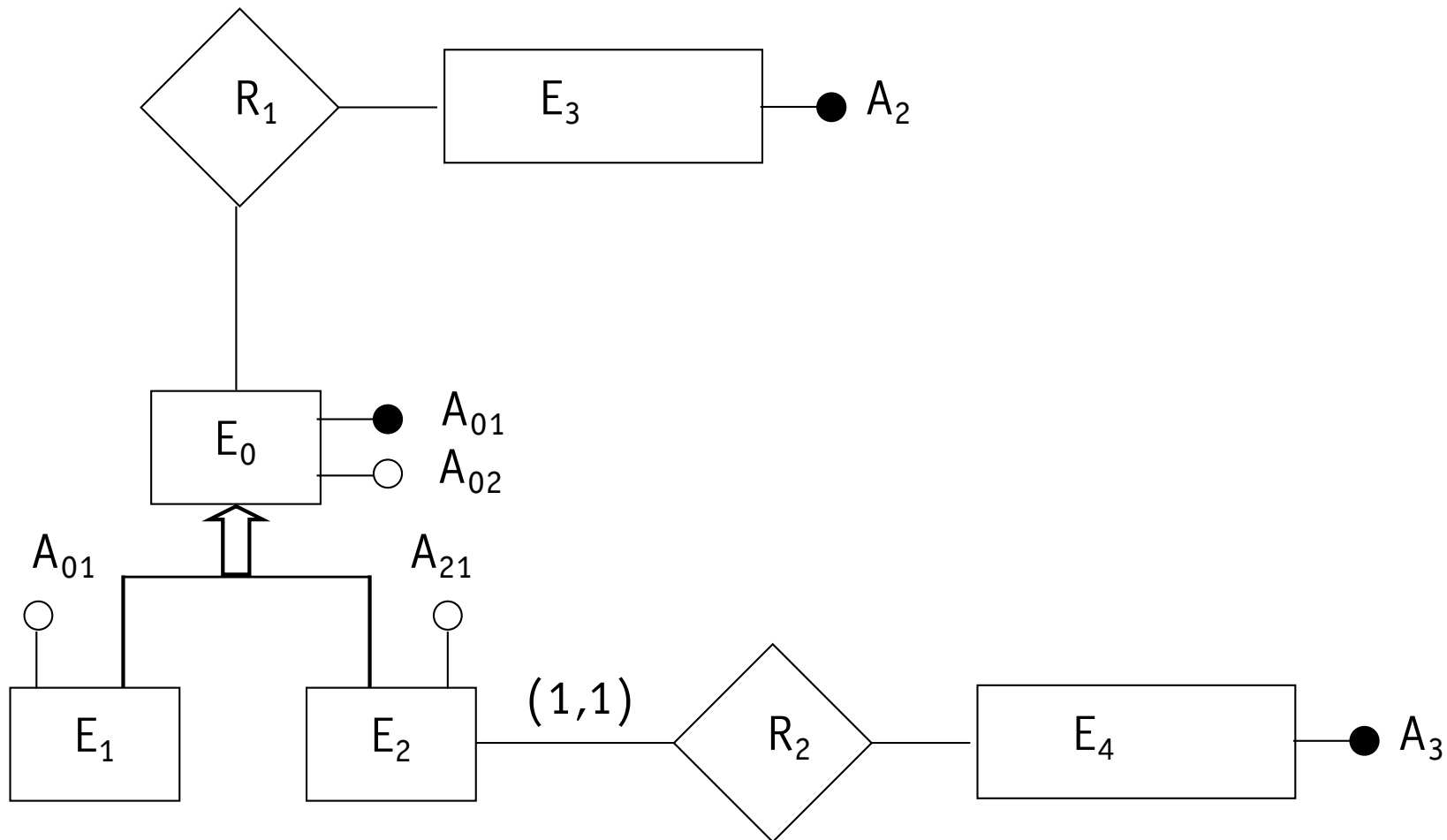
iii) Sostituzione della generalizzazione con associazioni



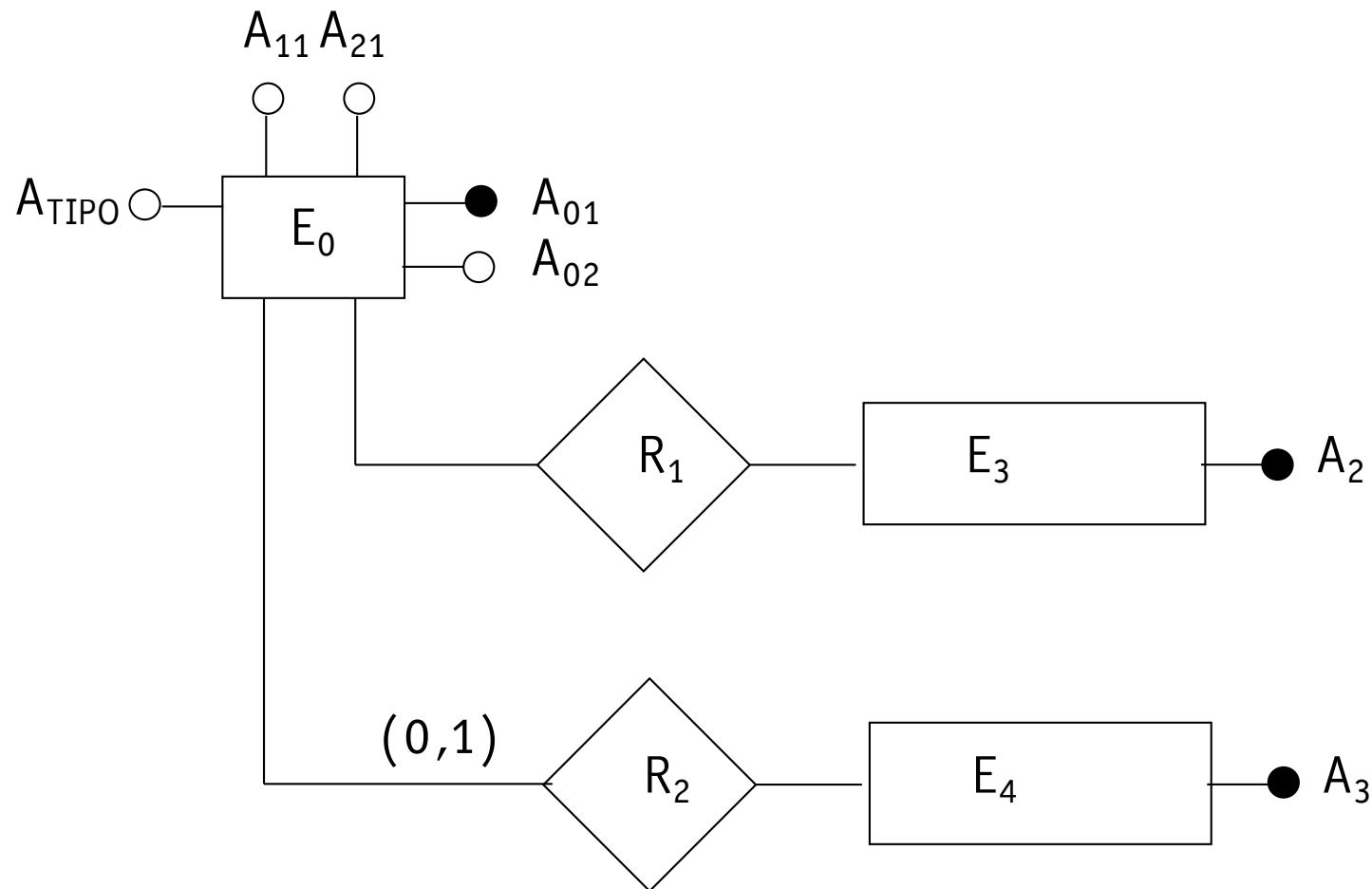
Considerazioni:

- È conveniente quando ci sono operazioni che accedono solo a istanze di E_1 (E_2) o E_0 , facendo distinzione negli accessi alle occorrenze delle entità figlie e di quella genitrice

Eliminazione delle generalizzazioni (caso generale)

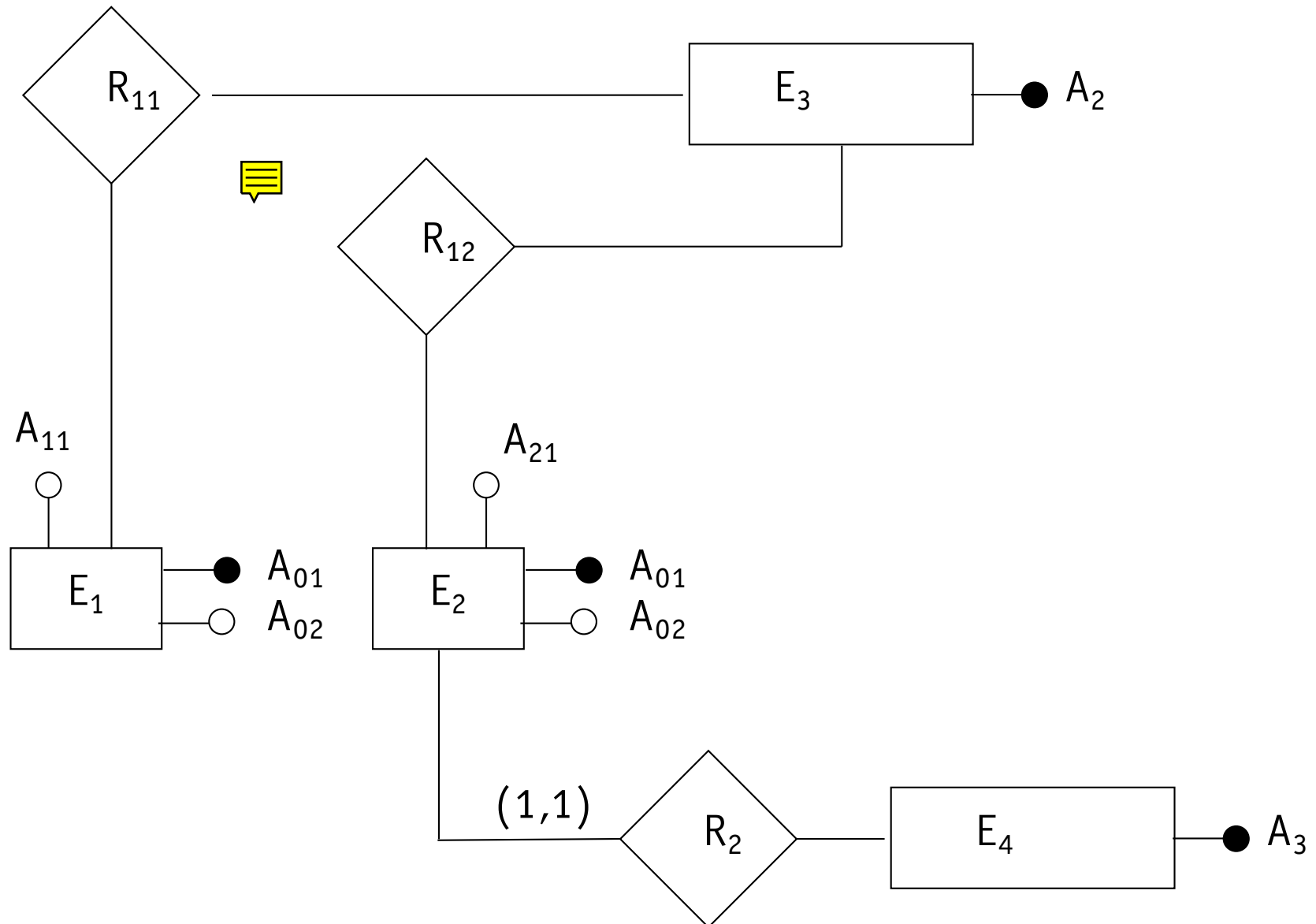


i) Accorpamento delle entità figlie nel genitore

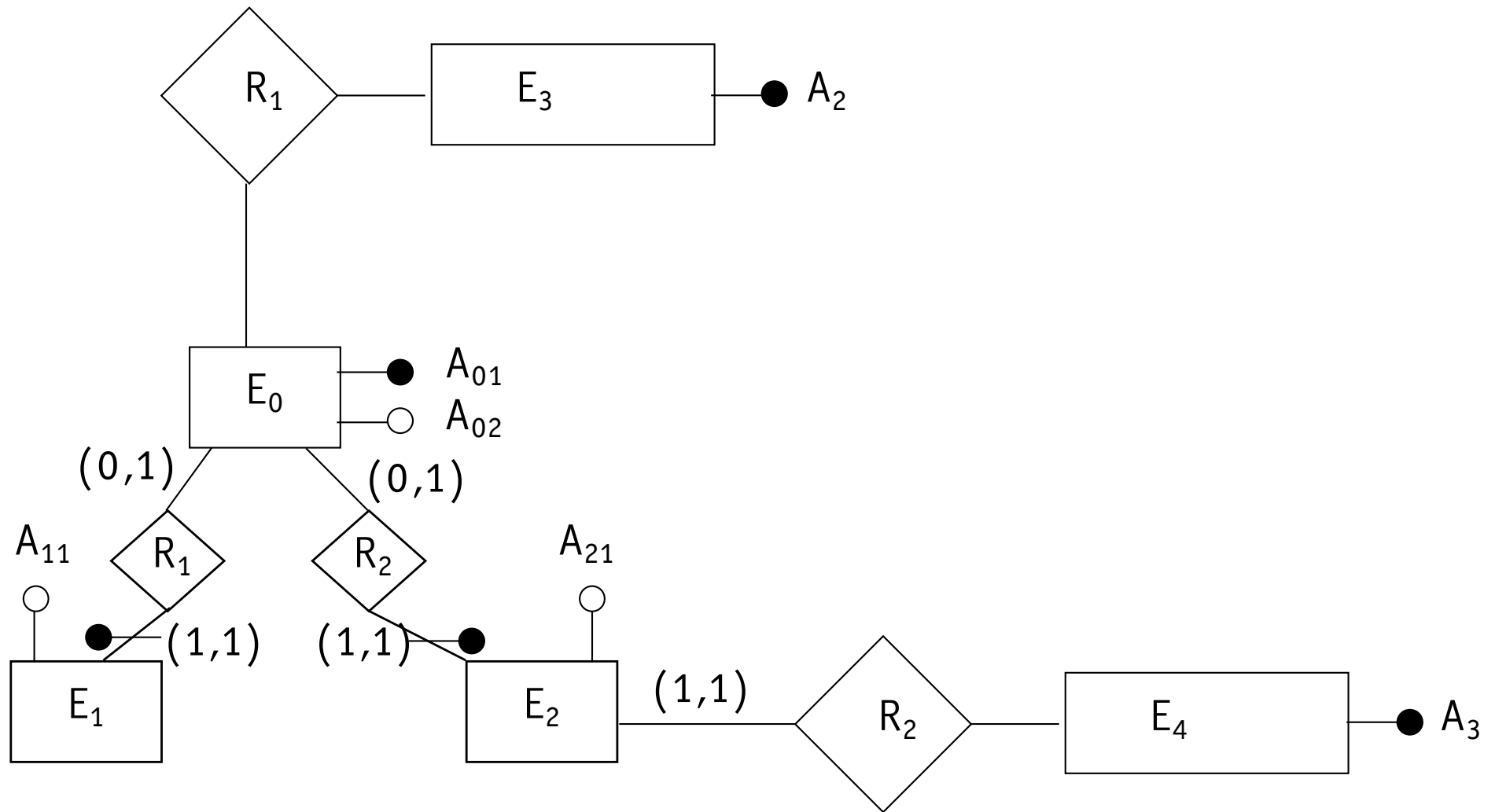


- Più vincolo che impone che un'occorrenza di E_0 partecipa a R_2 se e solo se è di tipo E_2

ii) Accorpamento del genitore nelle entità figlie



iii) Sostituzione della generalizzazione con associazioni



Riassumendo

- i) Accorpamento figli nel genitore: conviene se gli accessi non fanno distinzioni tra i figli
- ii) Accorpamento genitore nei figli: possibile solo se è totale; conviene se gli accessi alle entità figlie sono distinti
- iii) Sostituzione generalizzazione con associazioni: conviene se gli accessi ai figli sono separati dagli accessi al genitore

Sono anche possibili soluzioni “ibride”, utilizzate soprattutto in generalizzazioni a più livelli

3. Partizionamento/accorpamento di entità e associazioni

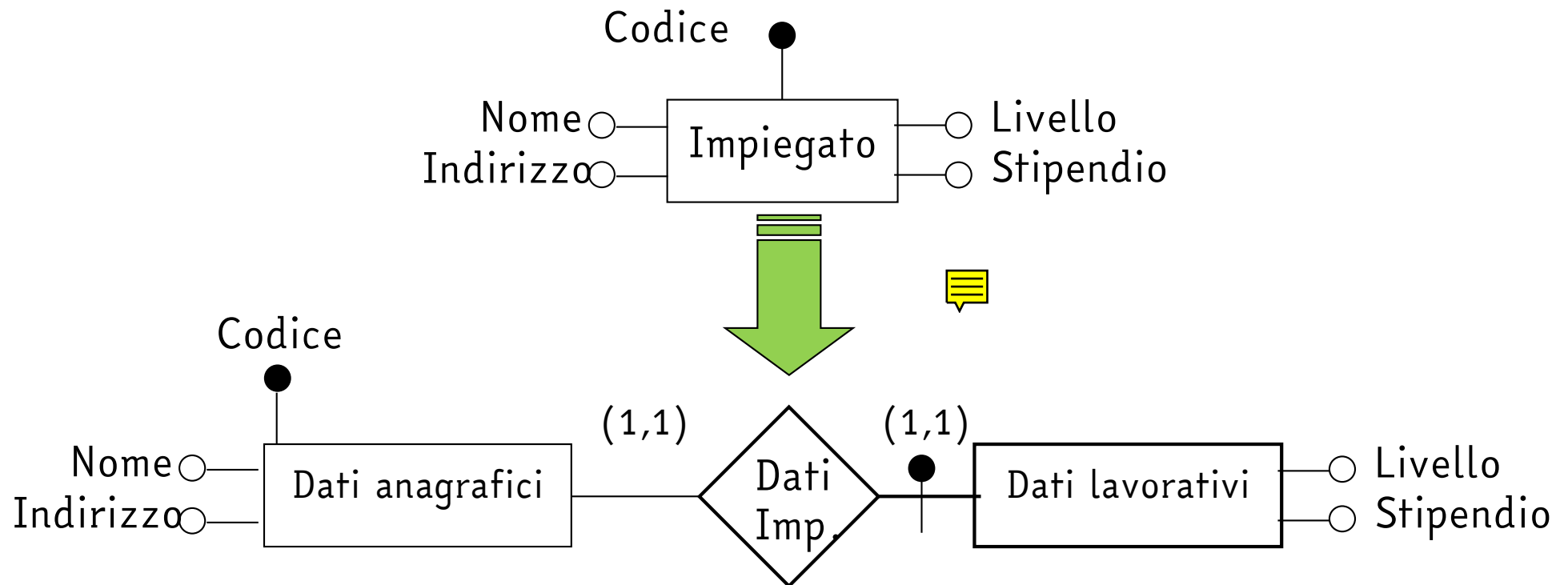
Il principio di base è il seguente.

Gli accessi si riducono:

- separando attributi di uno stesso concetto ai quali si accede in operazioni diverse
- accorpendo attributi di concetti diversi a cui si accede con le medesime operazioni

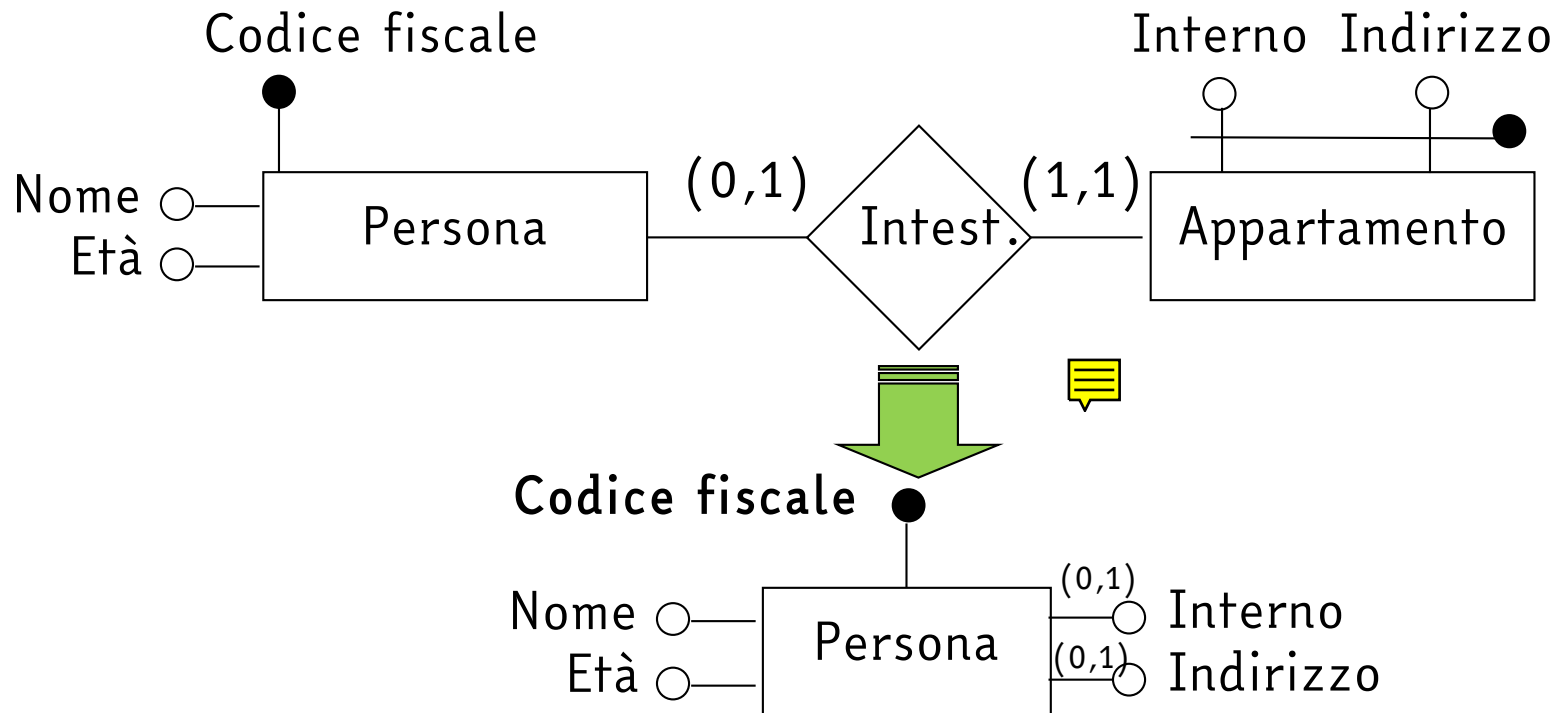
Spesso è possibile rimandare il problema del partizionamento/accorpamento alla fase della progettazione fisica (che non vedremo)

Partizionamento di entità



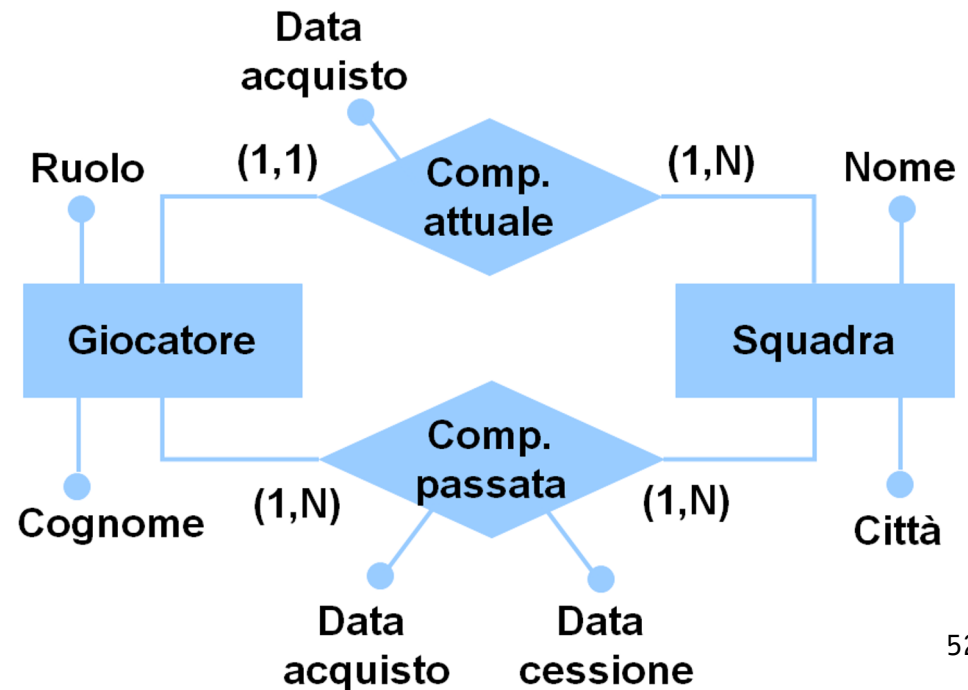
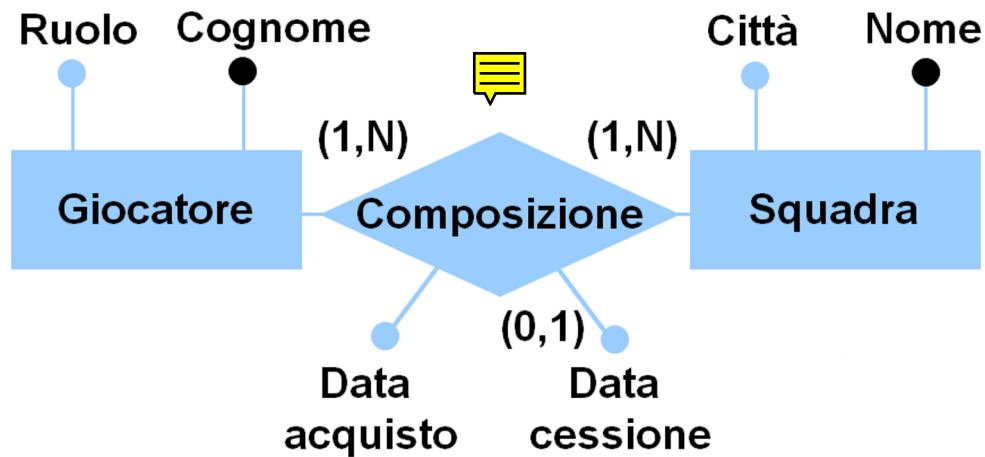
- L'effetto collaterale è la duplicazione di tutte le associazioni che coinvolgono l'entità partizionata

Accorpamento di entità



- Utile ad es. quando le transazioni che accedono a istanze di **Persona** necessitano sempre di informazioni relative anche all'appartamento
- Le cardinalità dell'associazione ci suggeriscono che possono esserci istanze con valori nulli per gli attributi **Interno** e **Indirizzo**, quindi aggiungiamo anche la regola aziendale **Interno è valorizzato ⇔ Indirizzo è valorizzato**
- Gli accorpamenti si fanno in genere su entità legate da associazioni 1 a 1, questo perché le associazioni molti a molti in caso di accorpamento generano schemi non normalizzati

Partizionamento di associazione

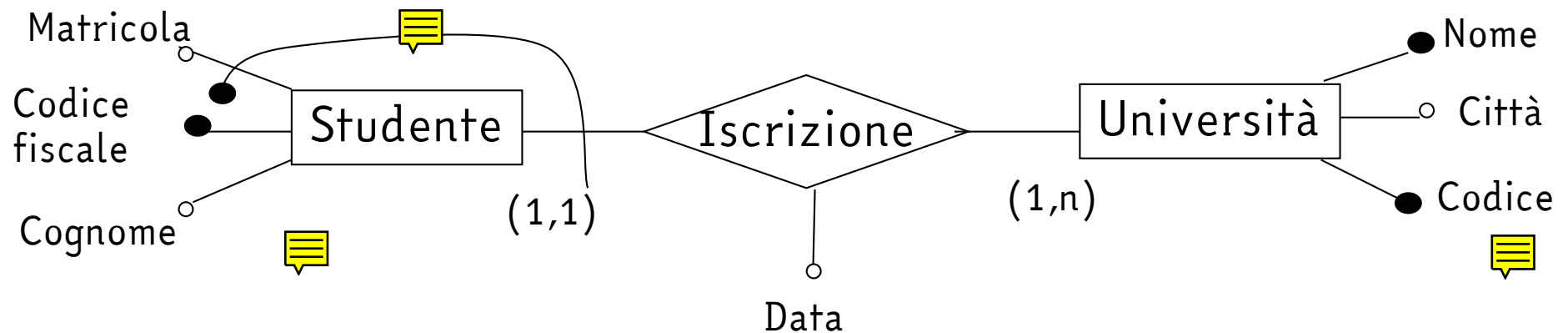


4. Scelta degli identificatori principali


- Operazione indispensabile per la traduzione nel modello relazionale
- Criteri
 - assenza di opzionalità
 - semplicità
 - utilizzo nelle operazioni più frequenti o importanti
- Se nessun identificatore rispetta questi requisiti si introducono nuovi attributi (per es. codici) contenenti valori speciali generati appositamente per questo scopo

4. Scelta degli identificatori principali

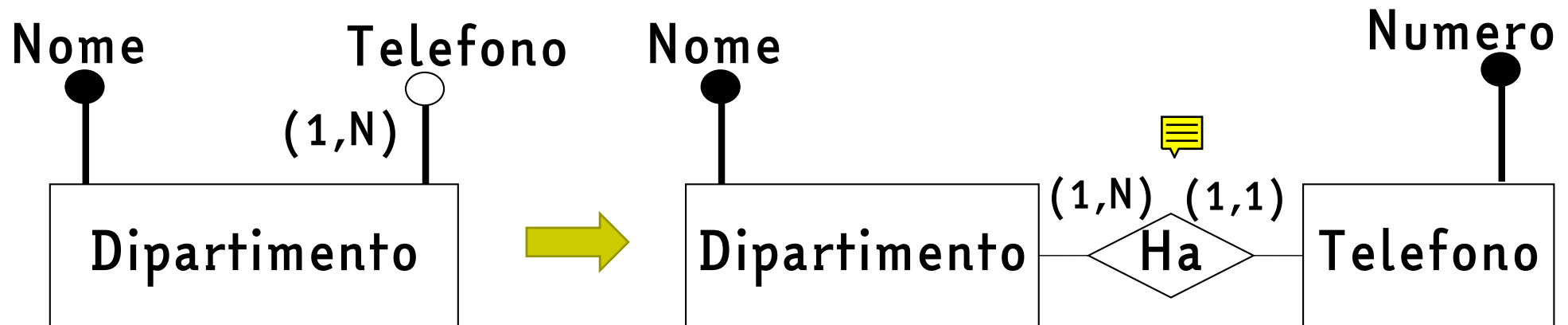
- Quali identificatori scegliereste?



4. Scelta degli identificatori principali

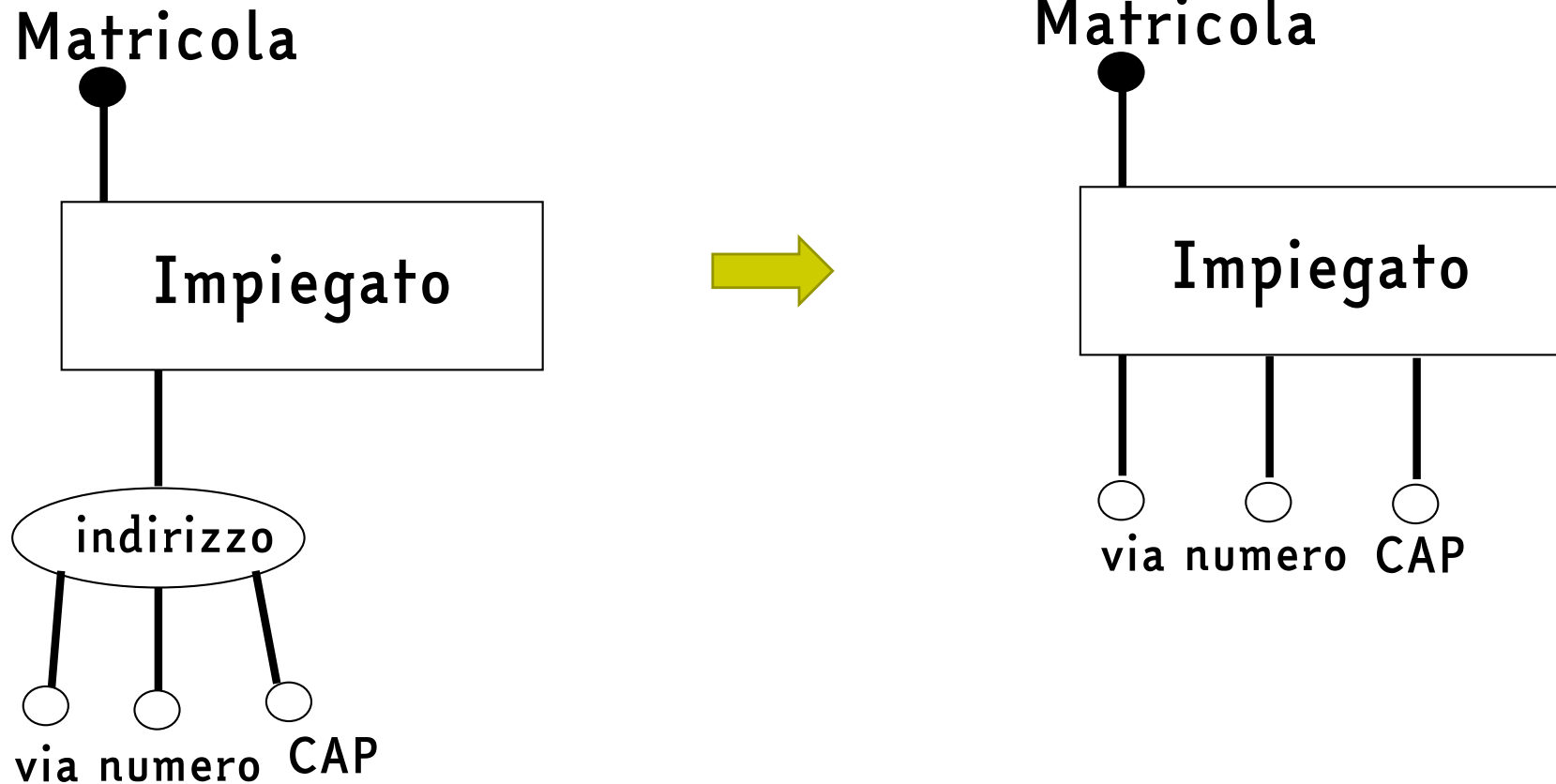
- Talvolta, per motivi di performance, cioè quando non è possibile avere un identificatore composto da pochi attributi o si ha un identificatore poco efficiente (per es. una stringa lunga), si crea un identificatore “surrogato” (che corrisponde di solito a un intero autoincrementante).
- Ad es. un identificatore surrogato di un'entità *Studente* potrebbe essere *ID* che contiene un numero univoco per ogni occorrenza.
- Attenzione perché tutti gli identificatori, anche quelli non scelti, dovranno conservare il vincolo  *unique* nello schema relazionale.

5. Attributi multivalore



Posso trasformare gli attributi multivalore, non rappresentabili direttamente in relazionale, reificando l'attributo e aggiungendo un'associazione

6. Attributi composti



Anche gli attributi composti non sono rappresentabili direttamente in relazionale e devono essere trasformati

Traduzione verso il modello logico

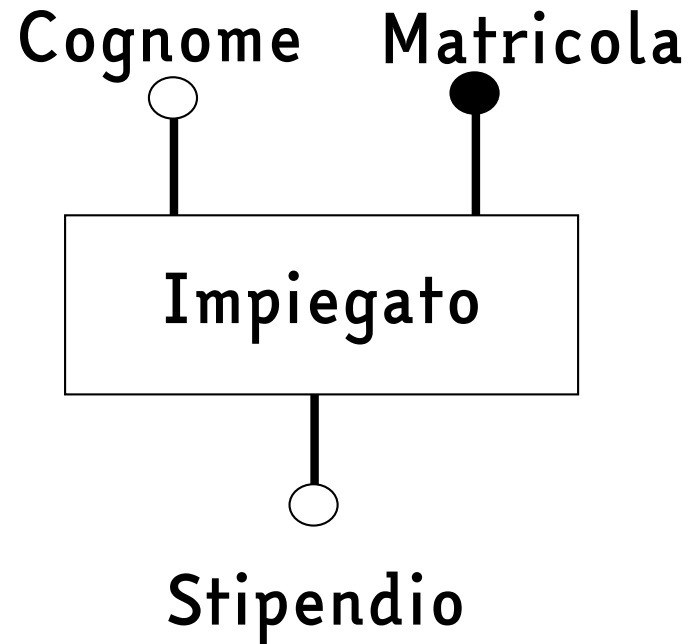
Traduzione verso il modello relazionale

idea di base

Le *entità* diventano relazioni con gli stessi attributi delle entità

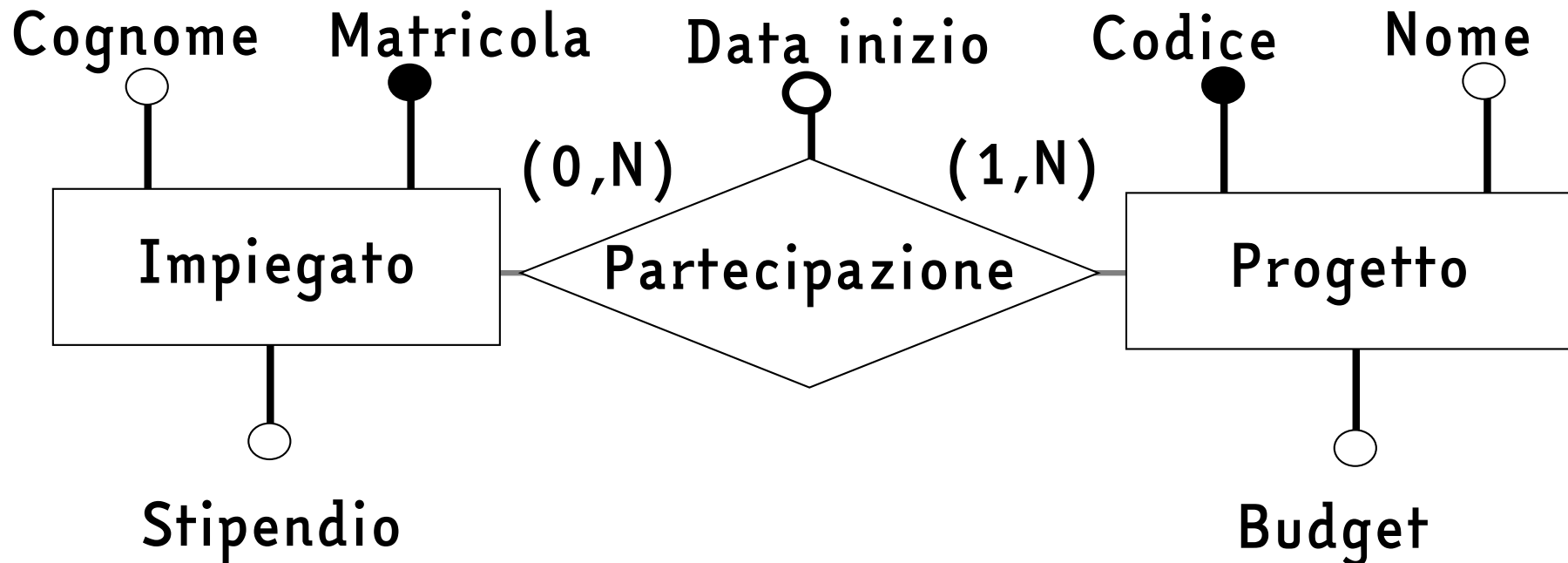
Le *associazioni* diventano relazioni con attributi delle associazioni + gli identificatori delle entità coinvolte

Entità



Impiegato(Matricola, Cognome, Stipendio)


Associazioni molti a molti



Impiegato(Matricola, Cognome, Stipendio)

Progetto(Codice, Nome, Budget)

Partecipazione(Matricola, Codice, DataInizio) 

- **Partecipazione**(Matricola)  referencia **Impiegato**(Matricola)
- **Partecipazione**(Codice) referencia **Progetto**(Codice)

Associazioni molti a molti

Ridenominazione attributi

Diamo nomi più espressivi agli attributi della chiave della relazione che rappresenta l'associazione

Impiegato(Matricola, Cognome, Stipendio)

Progetto(Codice, Nome, Budget)

~~Partecipazione(Matricola, Codice, DataInizio)~~

Partecipazione(Impiegato, Progetto, DataInizio)

Partecipazione(Impiegato) referenzia

Impiegato(Matricola)

Partecipazione(Progetto) referenzia

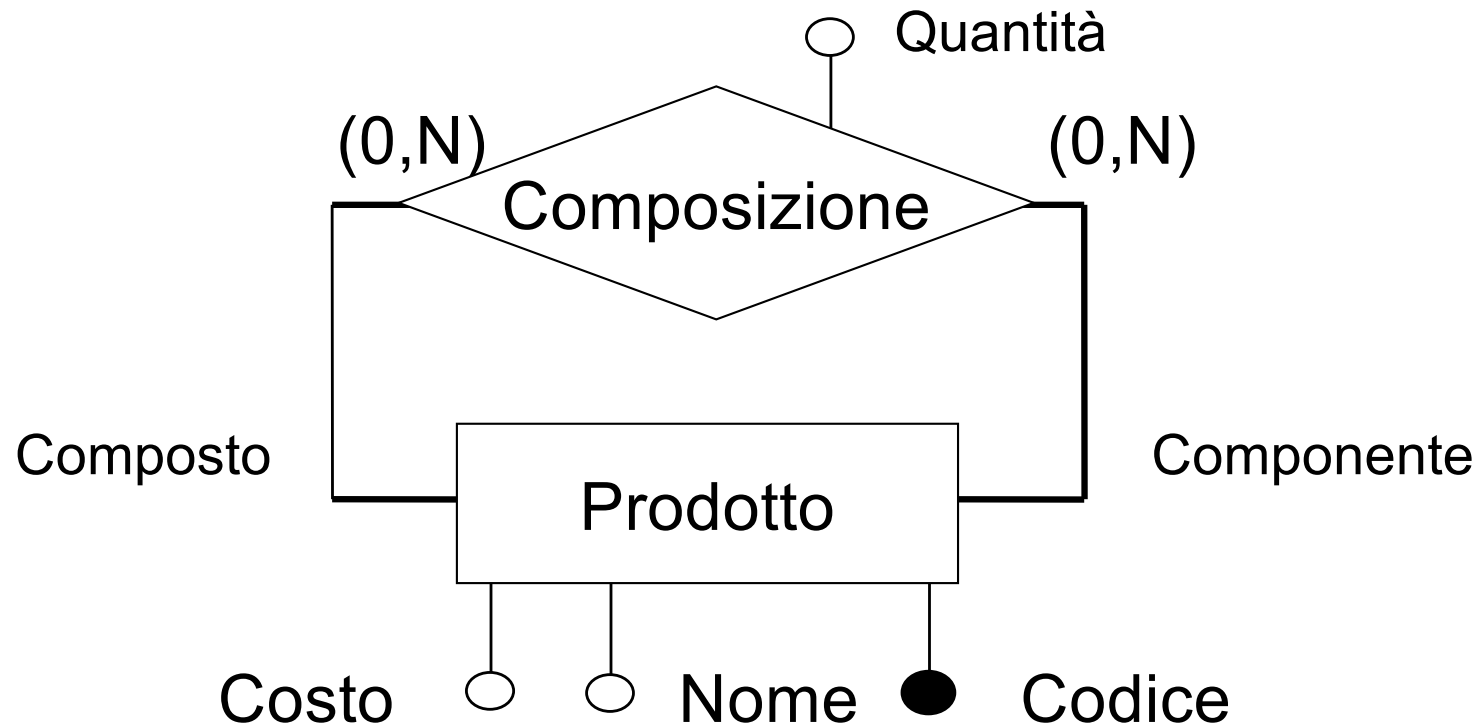
Progetto(Codice)

Associazioni molti a molti

Cardinalità minime

Nota: La traduzione non riesce a tener conto delle cardinalità minime delle associazioni molti a molti (se non con vincoli SQL di CHECK complessi e poco usati)

Associazioni ricorsive

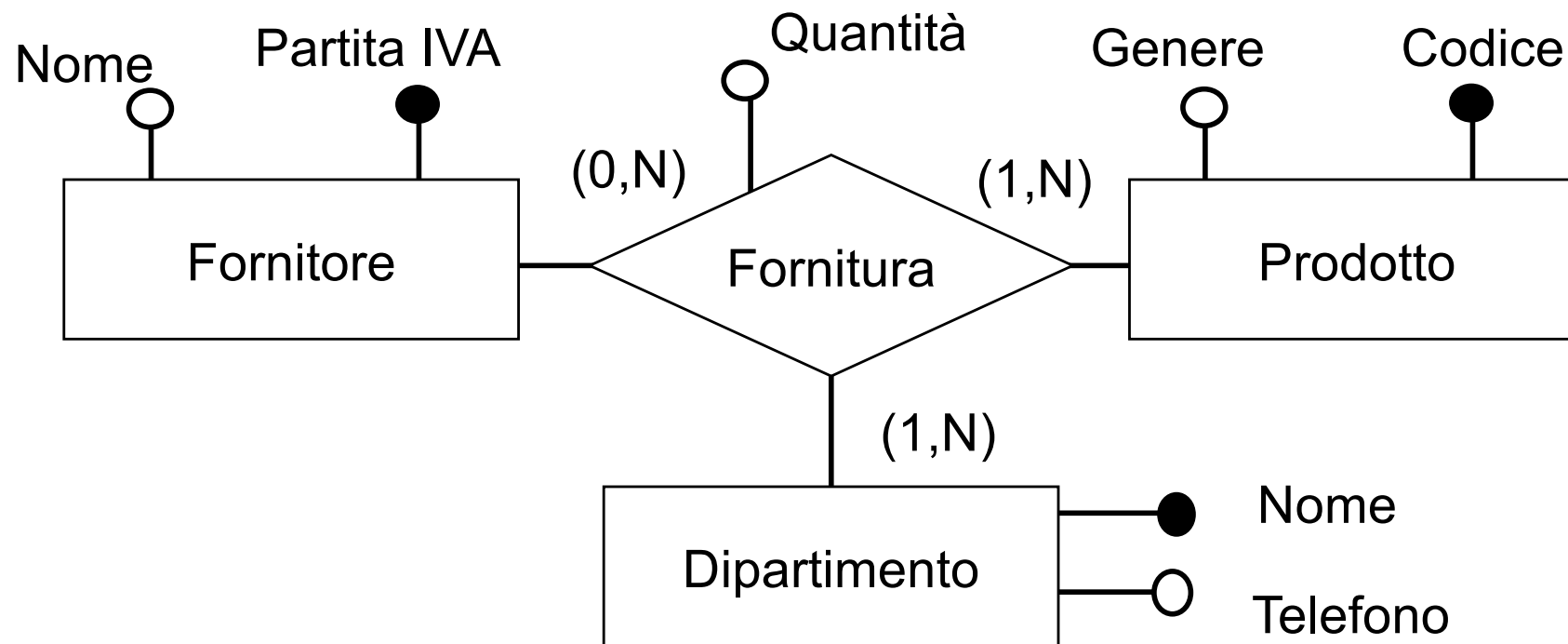


Prodotto(Codice, Nome, Costo)

Composizione(Composto, Componente, Quantità) 

- Composizione(Composto) e Composizione(Componente) 69
referenziano Prodotto(Codice)

Associazioni n-arie



Fornitore(PartitaIVA, Nome)

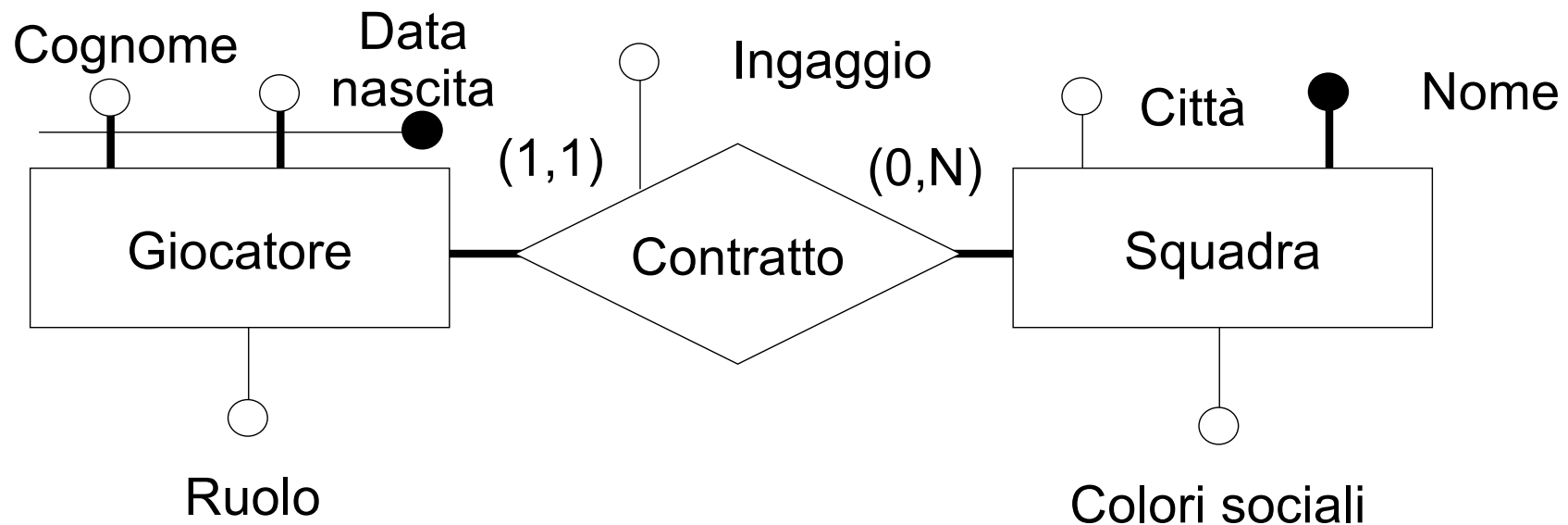
Prodotto(Codice, Genere)

Dipartimento(Nome, Telefono)

Fornitura(Fornitore, Prodotto, Dipartimento, Quantità)

- Fornitura(Fornitore) referencia Fornitore(PartitaIVA), Fornitura(Prodotto) referencia Prodotto(Codice), Fornitura(Dipartimento) referencia Dipartimento(Nome)

Associazioni uno a molti



Giocatore(Cognome, DataNascita, Ruolo)

Squadra(Nome, Città, ColoriSociali)

Contratto(CognGiocatore, DataNascG, Squadra, Ingaggio)

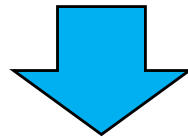
- È corretto? 

Associazioni uno a molti

Giocatore(Cognome, DataNascita, Ruolo)

Squadra(Nome, Città, ColoriSociali)

Contratto(CognomeG, DataNascitaG, Squadra, Ingaggio)



Giocatore(Cognome, DataNascita, Ruolo, Squadra,
Ingaggio)

Squadra(Nome, Città, ColoriSociali)

- Giocatore(Squadra) referencia Squadra(Nome)

Associazioni uno a molti

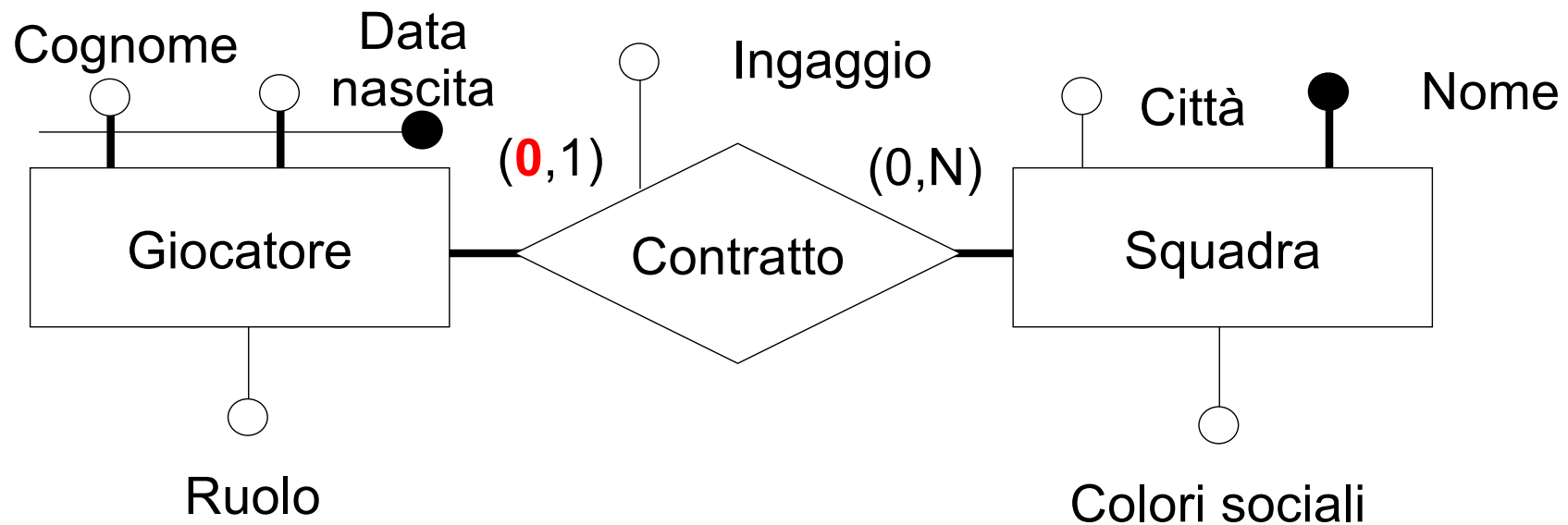
Cardinalità minime

Nota: La traduzione dell'associazione uno a molti riesce a rappresentare efficacemente la cardinalità minima della partecipazione che ha 1 come cardinalità massima:

0 : valore nullo ammesso

1 : valore nullo non ammesso

Associazioni uno a molti



Giocatore(Cognome, DataNascita, Ruolo, Squadra*, Ingaggio*)

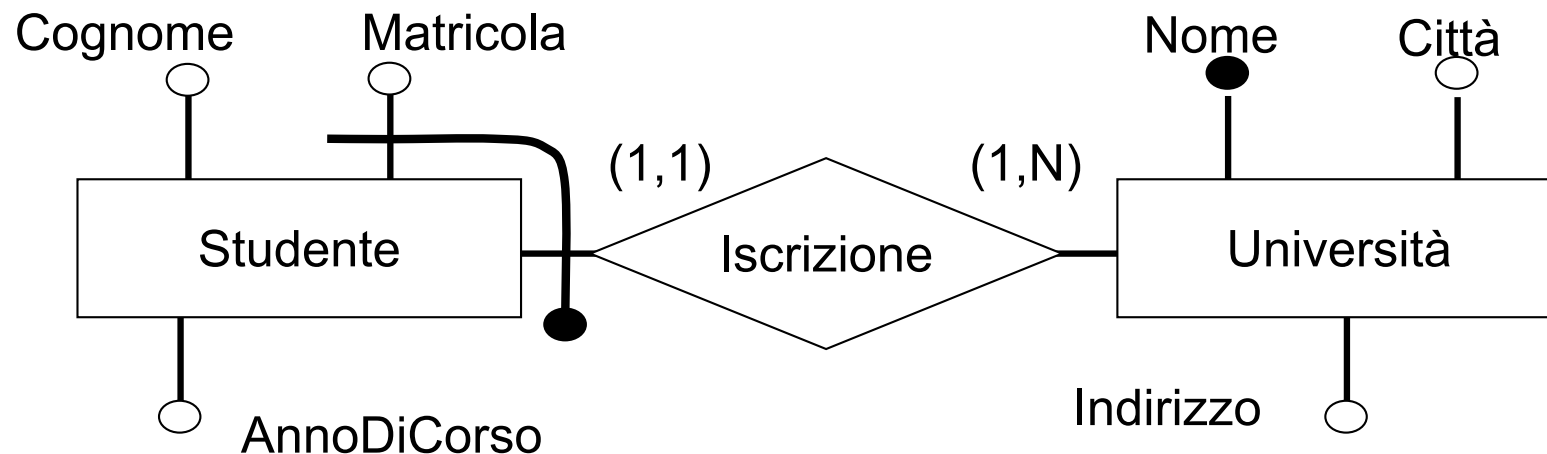
Squadra(Nome, Città, ColoriSociali)

- Se la cardinalità minima dell'associazione è 0, allora Squadra (e Ingaggio) in Giocatore devono ammettere valori nulli

Entità con identificazione esterna



L'identificazione esterna è sempre su un'associazione uno a molti o un'associazione uno a uno. Vediamo il caso, più comune, dell'associazione uno a molti



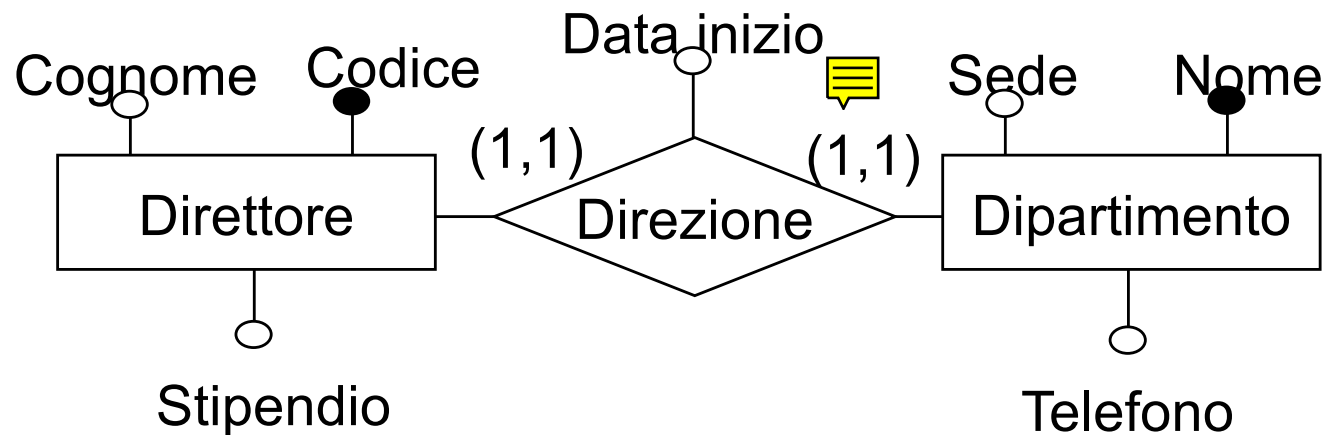
Studente(Matricola, Università, Cognome, AnnoDiCorso)

Università(Nome, Città, Indirizzo)

- Studente(Università) referencia Università(Nome)

Studente(Università) traduce Iscrizione; è in chiave primaria per tradurre l'identificatore esterno

Associazioni uno a uno



- Due possibilità simmetriche e ugualmente valide:

Direttore(Codice, Cognome, Stipendio, DipartimentoDiretto, InizioDirezione)

Dipartimento(Nome, Telefono, Sede)

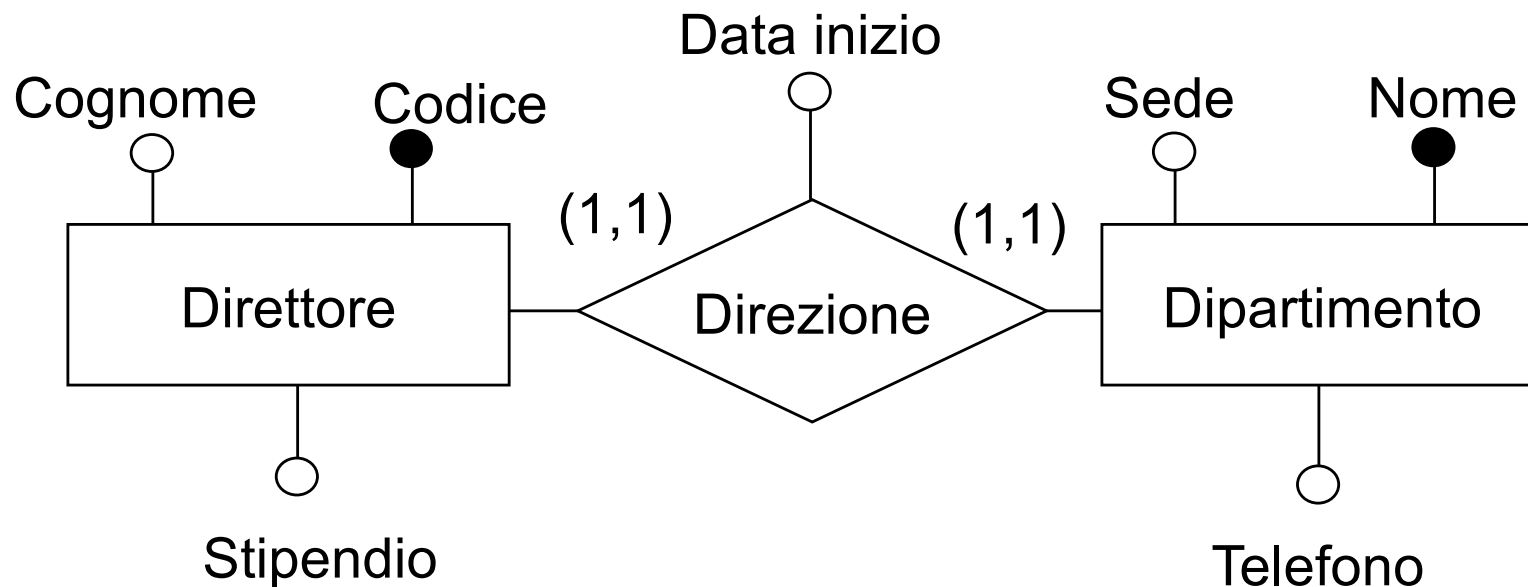
- Direttore(DipartimentoDiretto) referencia Dipartimento(Nome)
- (vincolo unique su Direttore(DipartimentoDiretto))


Direttore(Codice, Cognome, Stipendio)

Dipartimento(Nome, Telefono, Sede, Direttore, InizioDirezione)

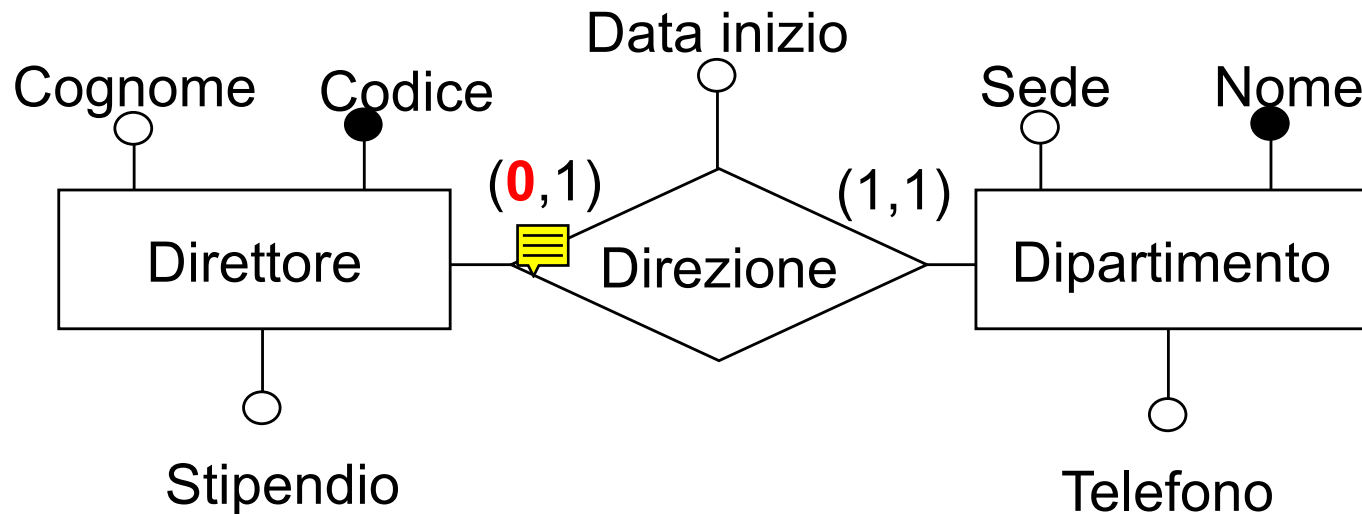
- Dipartimento(Direttore) referencia Direttore(Codice)
- (vincolo unique su Dipartimento(Direttore))

Associazioni uno a uno



- Una terza possibilità: rappresentare tutti i concetti in una singola relazione 
- Svantaggi: nello schema ER le due entità sono state rappresentate separatamente per qualche ragione, quindi è meglio tenere le relative relazioni separate anche nel modello logico

Associazioni uno a uno



- Traduciamo l'associazione mediante attributi della relazione Dipartimento perché partecipa all'associazione con cardinalità (1,1):

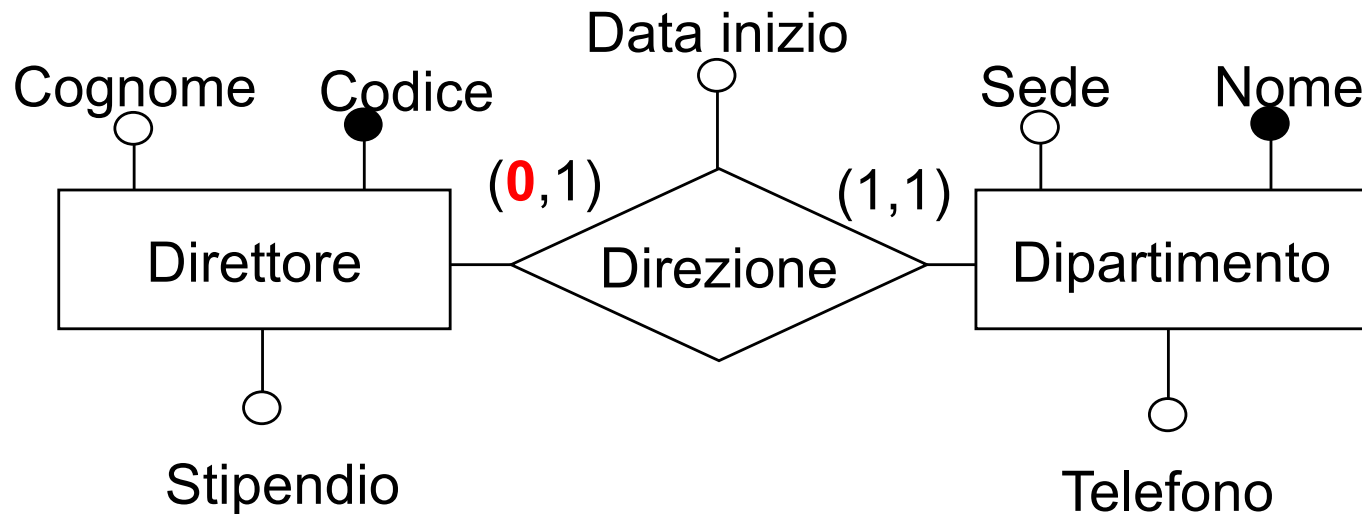
Direttore(Codice, Cognome, Stipendio)

Dipartimento(Nome, Telefono, Sede, Direttore, InizioDirezione)

Dipartimento(Direttore) referencia Direttore(Codice)

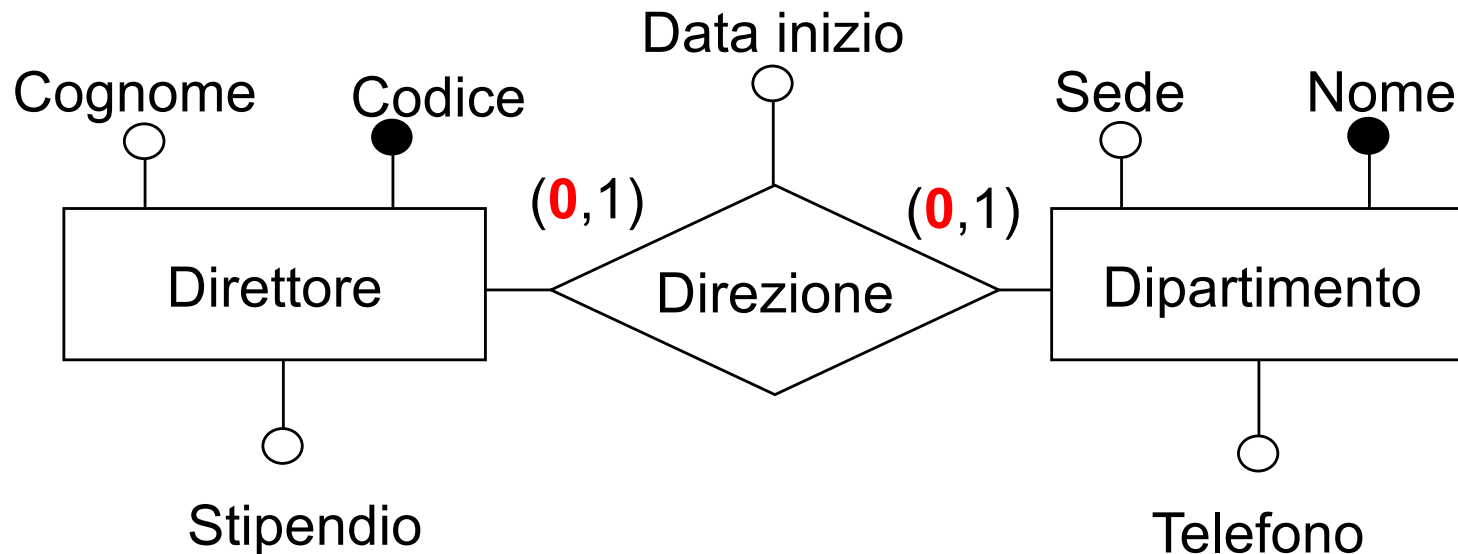
Vincolo unique su Dipartimento(Direttore)

Associazioni uno a uno



- L'alternativa in cui l'associazione Direzione viene rappresentata nella relazione Direttore è peggiore:
Direttore(Codice, Cognome, Stipendio, Dipartimento*, InizioDirezione*)
Dipartimento(Nome, Telefono, Sede)
- Infatti, poiché l'associazione tra Direttore e Direzione ha cardinalità (0,1), nella relazione Direttore per il nome del dipartimento diretto sono possibili valori nulli

Associazioni uno a uno

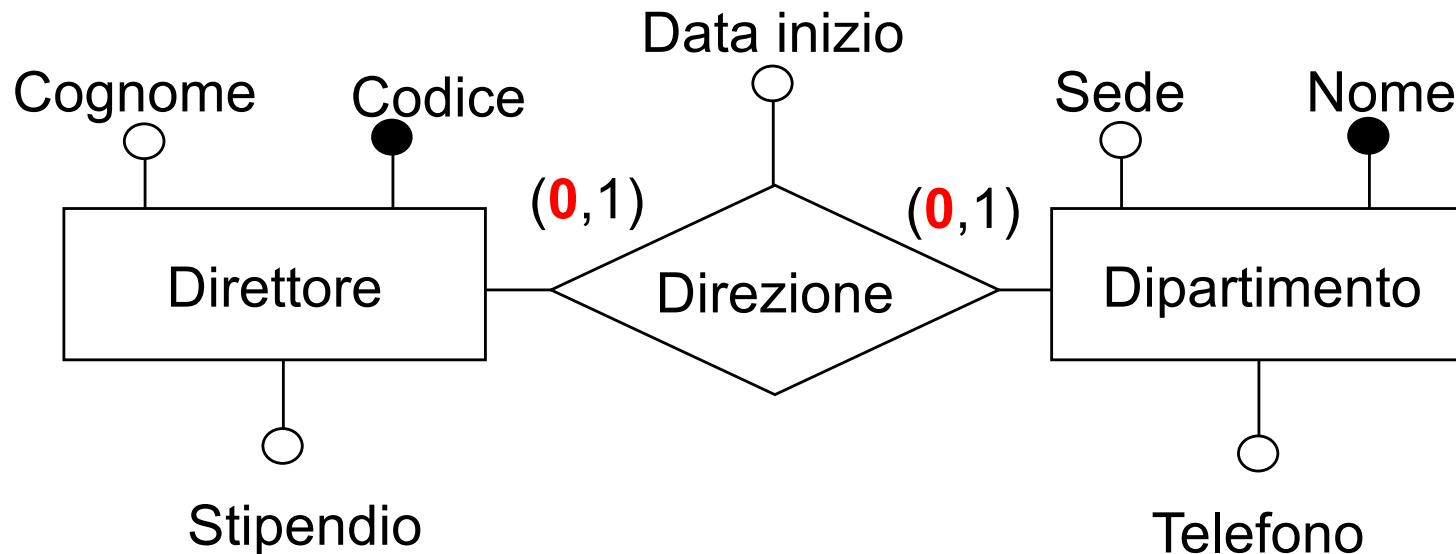


Tre opzioni:

1. Fondere l'entità **Direttore** e l'associazione **Direzione** in una relazione
2. Fondere l'entità **Dipartimento** e l'associazione **Direzione** in una relazione

1. e 2. sono simili al caso precedente

Associazioni uno a uno



3. Tradurre l'associazione Direzione con una relazione:

Direttore(Codice, Cognome, Stipendio)

Dipartimento(Nome, Telefono, Sede)

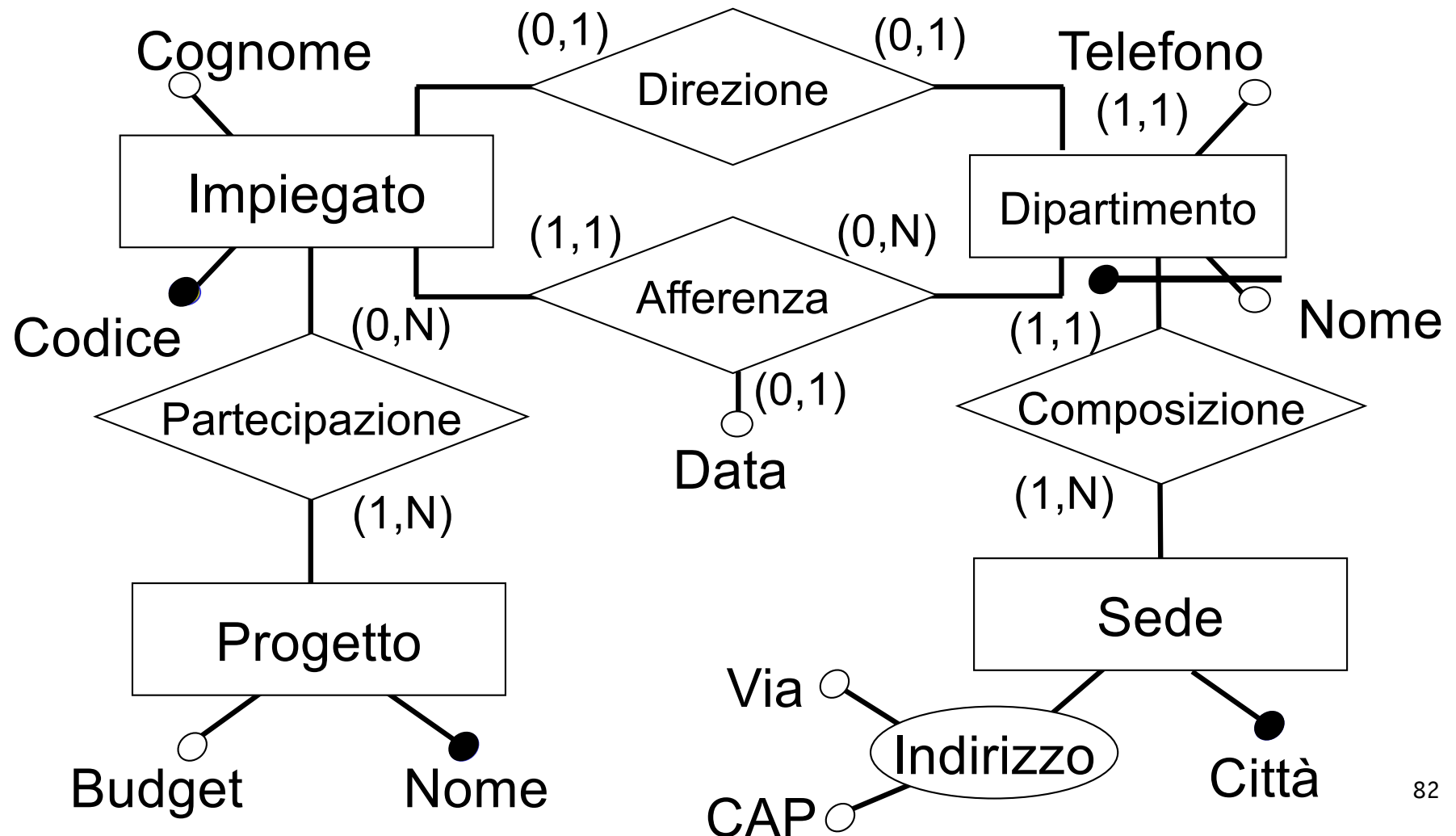
Direzione(Direttore, Dipartimento, DataInizioDirezione)

- Direzione(Direttore) referencia Direttore(Codice)
- Direzione(Dipartimento) referencia Dipartimento(Nome)
- Vincolo unique su Direzione(Dipartimento)

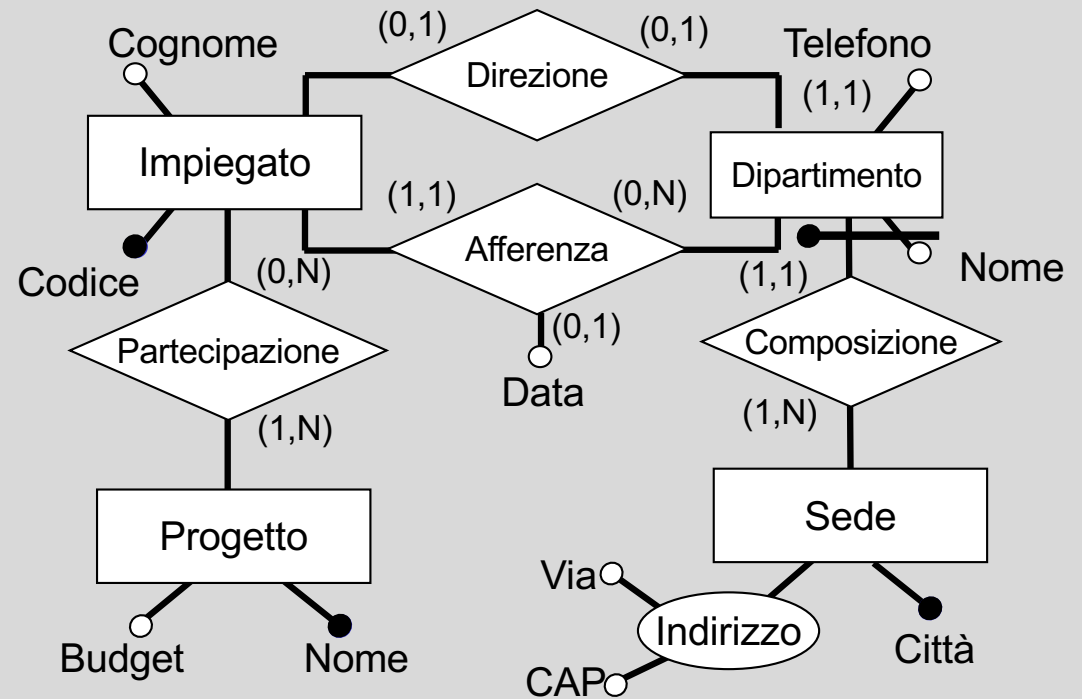
Vantaggio rispetto alle opzioni 1. e 2.: non ci sono i valori nulli

Esercizio

Traduciamo in relazionale (suggerimento: partiamo da Sede)



Esercizio



Sede(Città, Via, CAP)

Dipartimento(Nome, Sede, Telefono, Direttore*)

Impiegato(Codice, Cognome, DipartimentoNome, DipartimentoSede, Data*)

Progetto(Nome, Budget)

Partecipazione(Impiegato, Progetto)

Vincoli:

- Impiegato(DipartimentoNome, DipartimentoSede) referencia Dipartimento(Nome, Sede),
- Dipartimento(Direttore) referencia Impiegato(Codice),
- Dipartimento(Sede) referencia Sede(Città),
- Partecipazione(Impiegato) referencia Impiegato(Codice),
- Partecipazione(Progetto) referencia Progetto(Nome)
- Vincolo unique su Dipartimento(Direttore)