

Progettazione logica

- Obiettivo della fase di progettazione logica è pervenire, a partire dallo schema concettuale, a uno schema logico che lo rappresenti in **modo fedele** e che sia, al tempo stesso, “**efficiente**”
- **L’efficienza** è legata alle **prestazioni**, ma poiché queste non sono valutabili precisamente a livello concettuale e logico si ricorre a degli indicatori semplificati

Progettazione logica

La progettazione logica può articolarsi in due fasi principali:

- **Ristrutturazione**: eliminazione dallo schema E/R di tutti i costrutti che non possono essere direttamente rappresentati nel modello logico target (relazionale nel nostro caso):
 - Eliminazione degli attributi multivalore
 - Eliminazione delle generalizzazioni
 - Partizionamento/accorpamento di entità e relazioni
 - Scelta degli identificatori principali
- **Traduzione**: i costrutti residui si trasformano in elementi del modello relazionale

Fase di ristrutturazione

Serve a **semplificare la traduzione** e a “**ottimizzare**” le prestazioni

- Per confrontare tra loro diverse alternative bisogna conoscere, almeno in maniera approssimativa, il “**carico di lavoro**”, ovvero:
 - Le **principali operazioni** che il DB dovrà supportare
 - I “**volumi**” dei dati in gioco
- Gli indicatori che deriviamo considerano due aspetti
 - **spazio**: numero di istanze previste
 - **tempo**: numero di istanze (di entità e associazioni) visitate durante un'operazione

Schema di riferimento

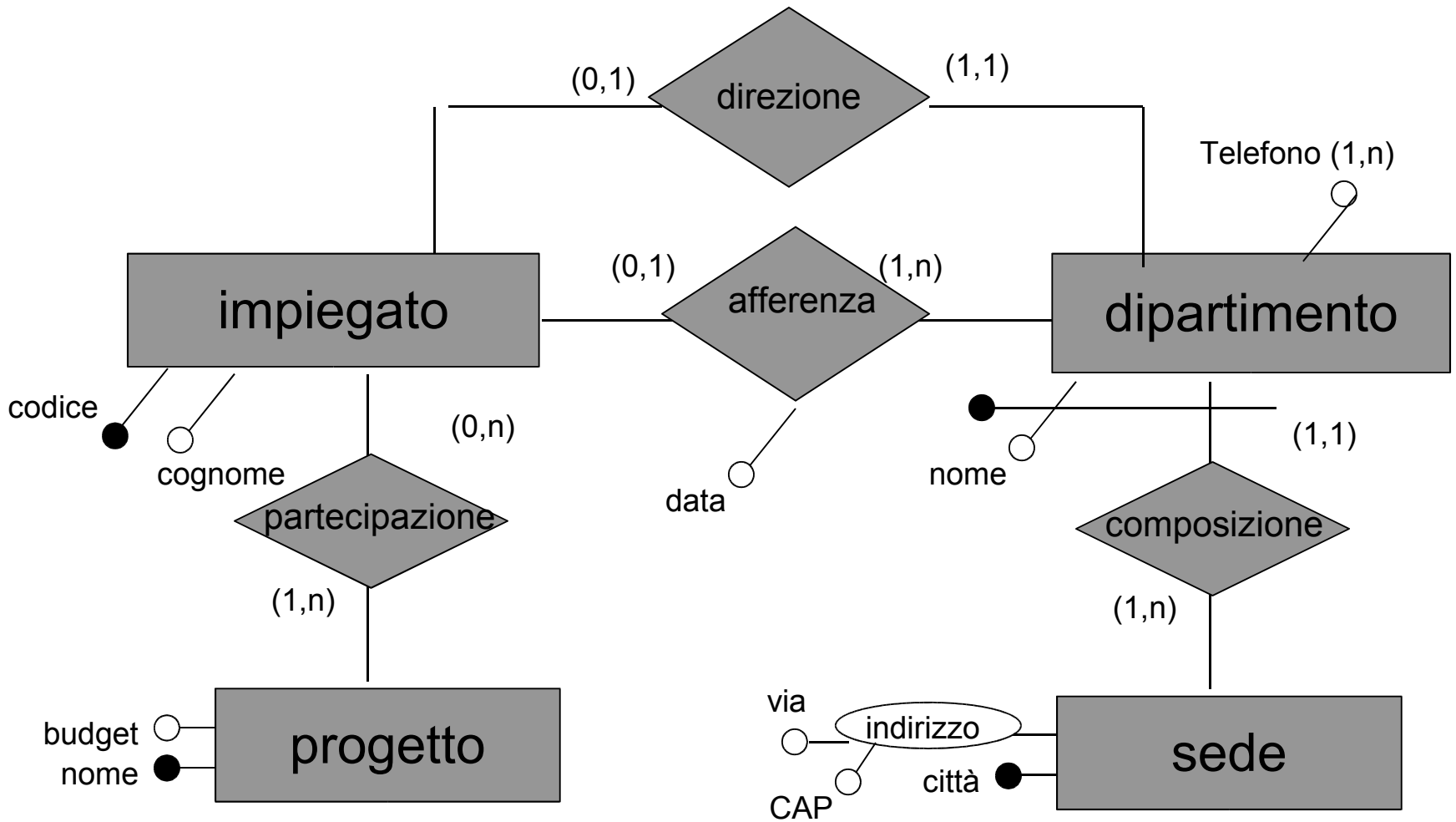


Tavola dei volumi

- Specifica il numero stimato di istanze per ogni entità (E) e relazione (R) dello schema
- I valori sono necessariamente approssimati, ma indicativi

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

Esempio di valutazione di costo

trova tutti i dati di un impiegato, del dipartimento ne quale lavora e dei progetti ai quali partecipa

- Si costruisce una **tavola degli accessi** basata su uno schema di navigazione
- Lo **schema di navigazione** è la parte dello schema E/R interessata dall'operazione, estesa con delle frecce che indicano in che modo l'operazione "naviga" i dati

Schema di navigazione

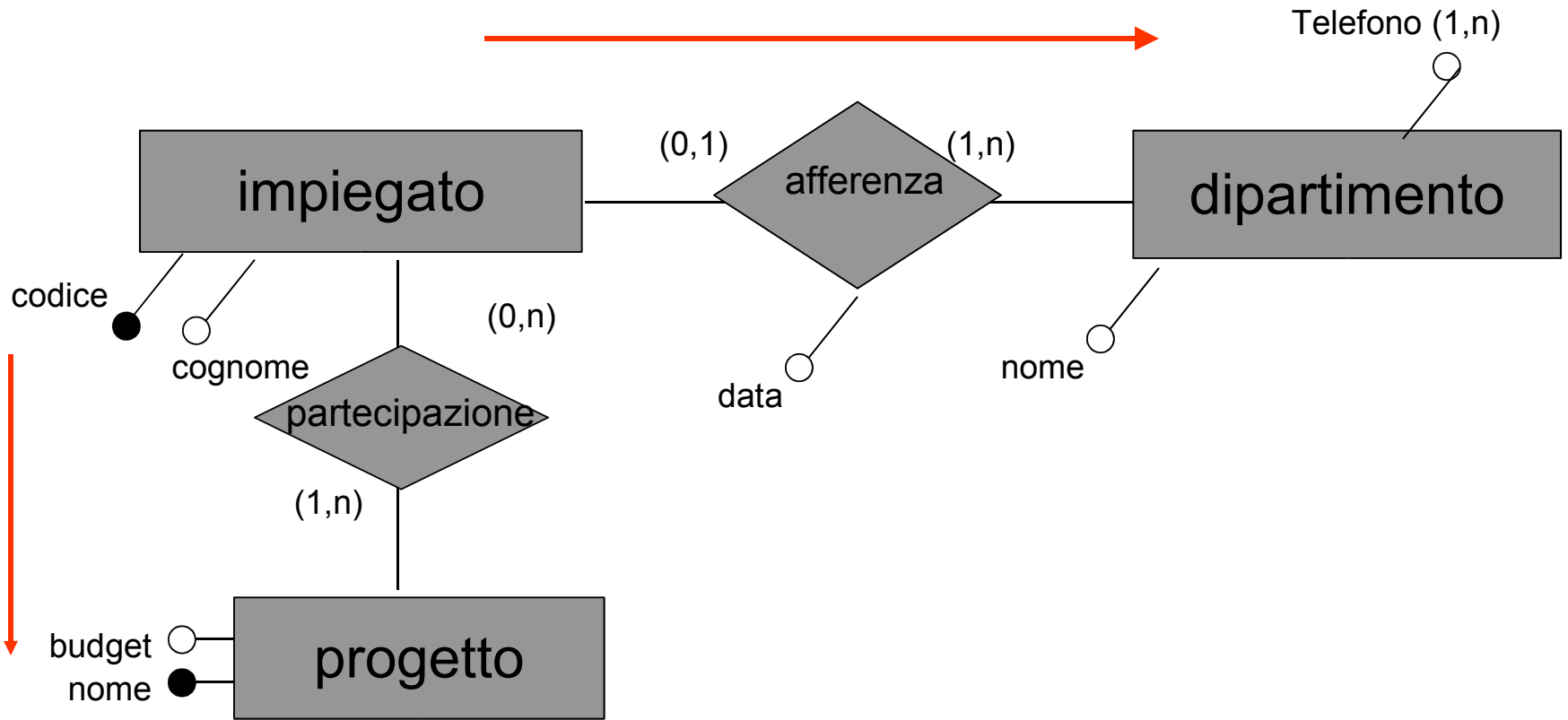


Tavola degli accessi

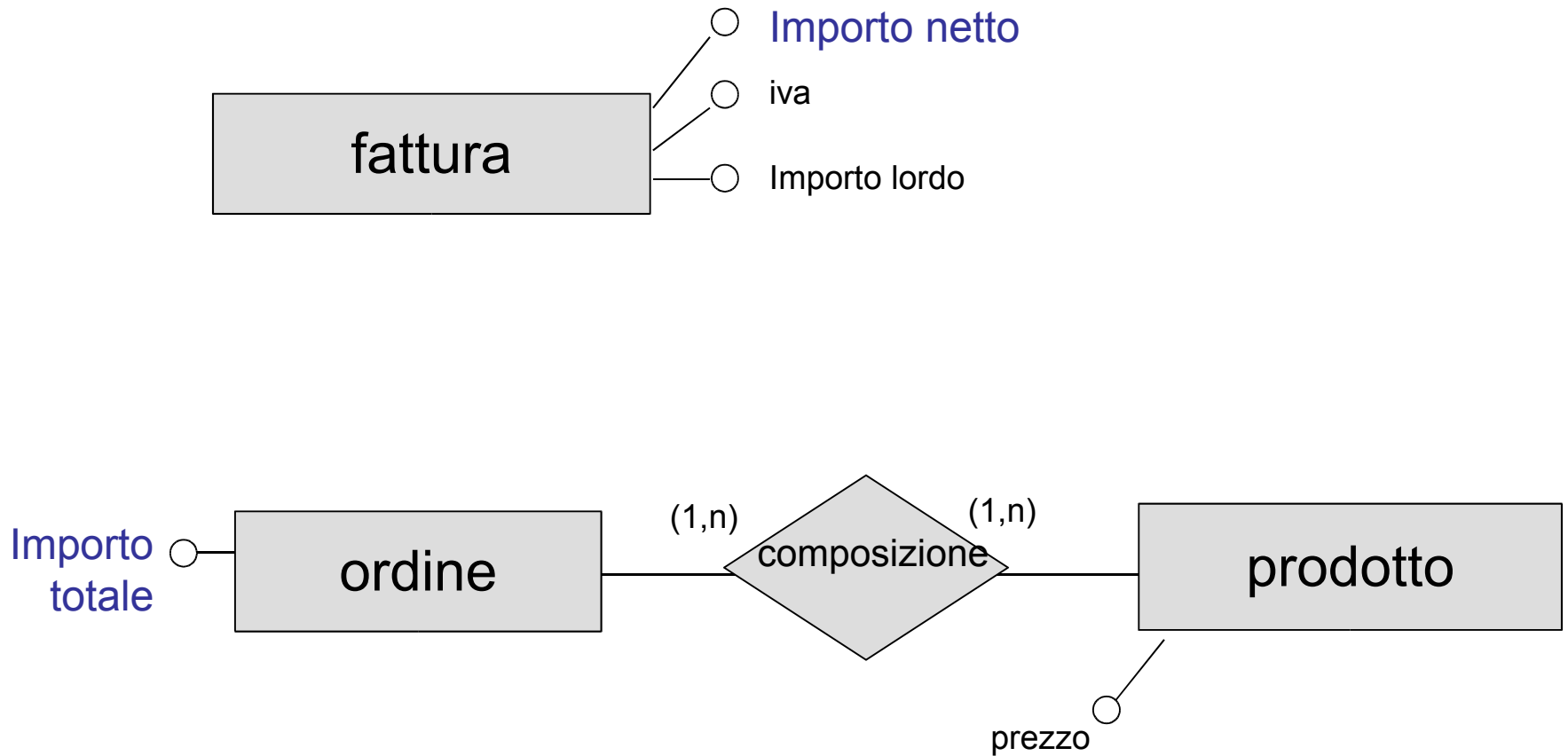
- Per ogni entità e associazione interessata dall'operazione, riporta il **numero di istanze interessate**, e il **tipo di accesso** (L: lettura; S: scrittura)
- Il numero delle istanze si ricava dalla tavola dei volumi mediante semplici operazioni (ad es: in media ogni impiegato partecipa a $6000/2000 = 3$ progetti)

Concetto	Costrutto	Accessi	Tipo
Impiegato	E	1	L
Afferenza	R	1	L
Dipartimento	E	1	L
Partecipazione	R	3	L
Progetto	E	3	L

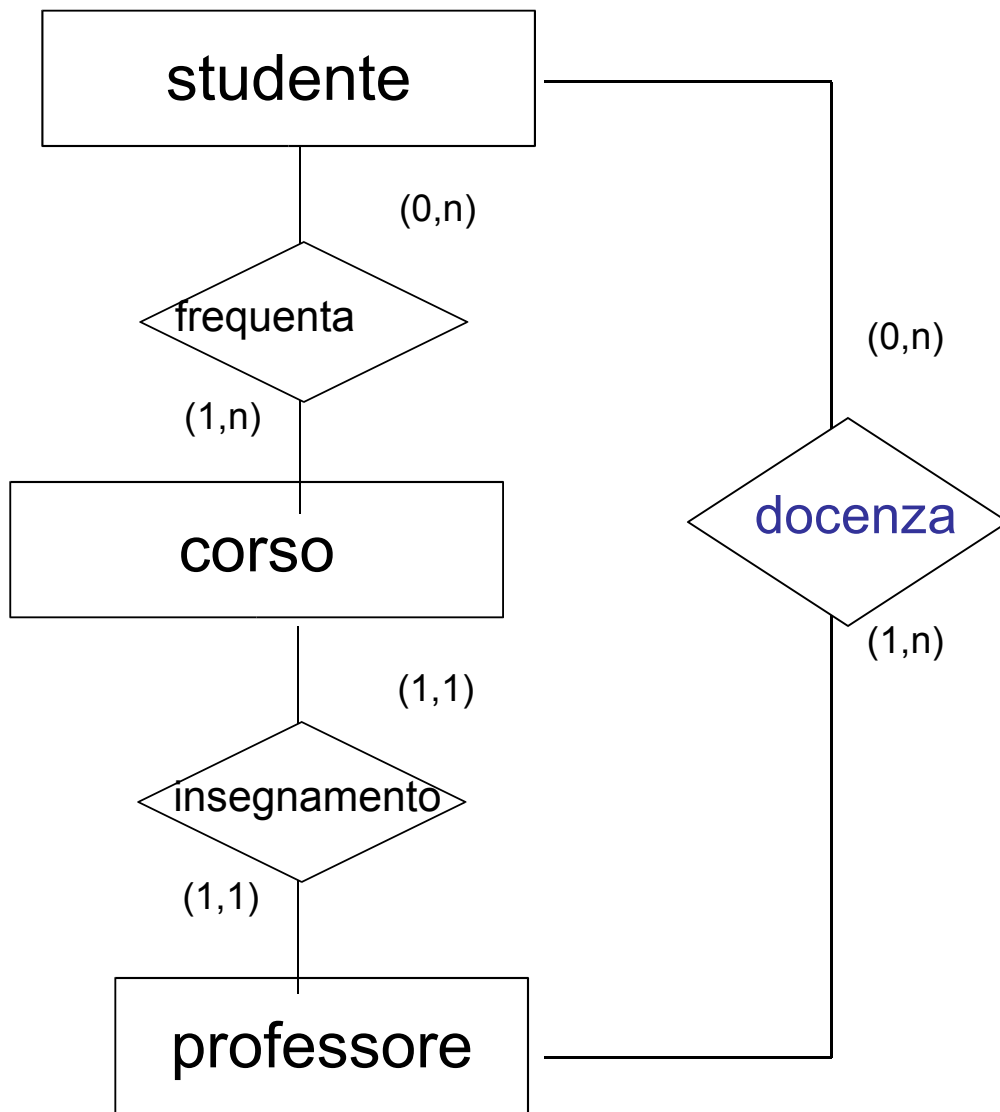
Analisi delle ridondanze

- 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 (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
 - si occupa maggior spazio
- Le possibili ridondanza riguardano
 - Attributi derivabili da altri attributi
 - Relazioni derivabili dalla composizione di altre relazioni (presenza di cicli)

Attributi derivabili



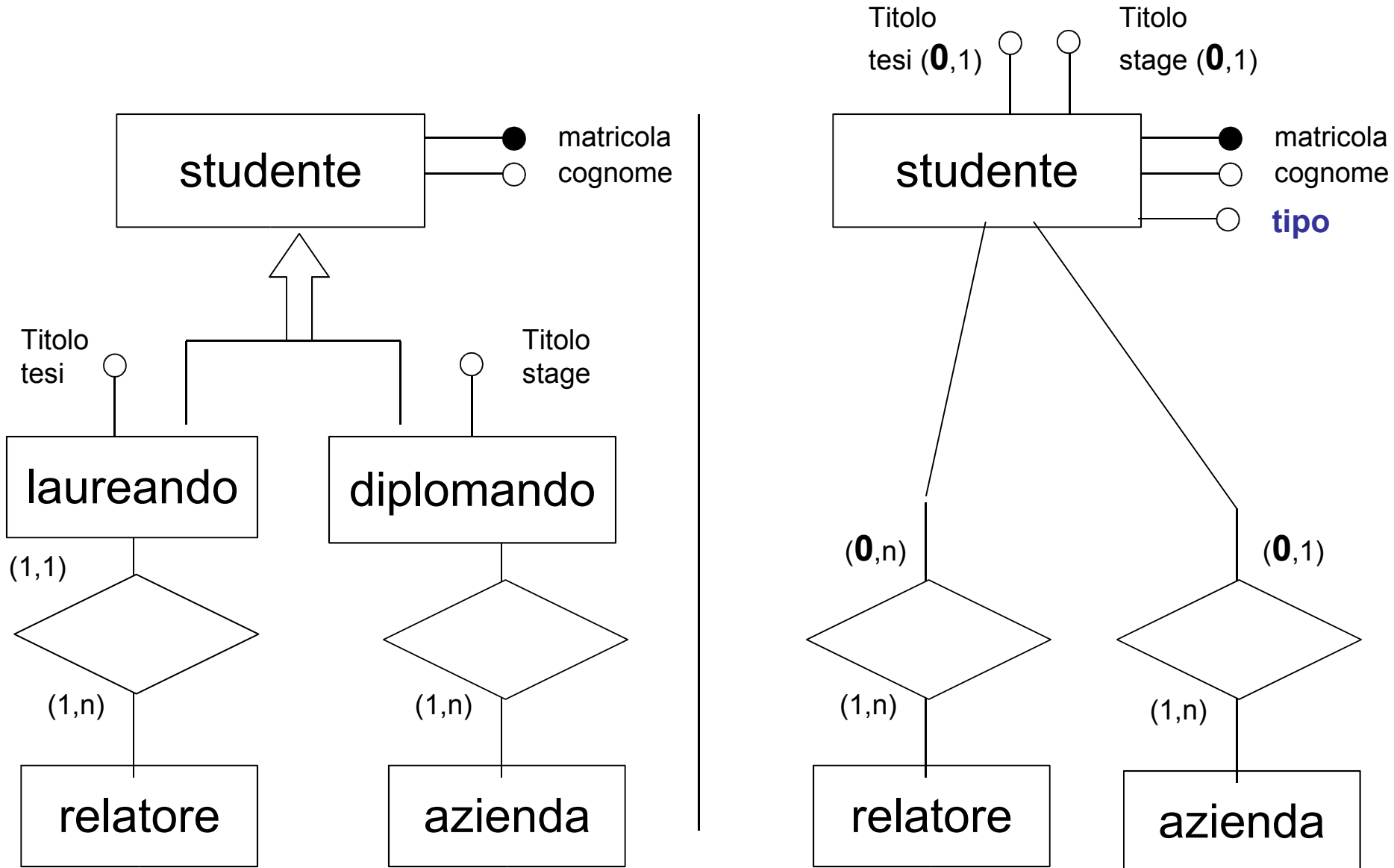
Associazioni ridondanti



Gerarchie

- Il modello relazionale non può rappresentare direttamente le generalizzazioni
- Entità e relazioni 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 relazioni

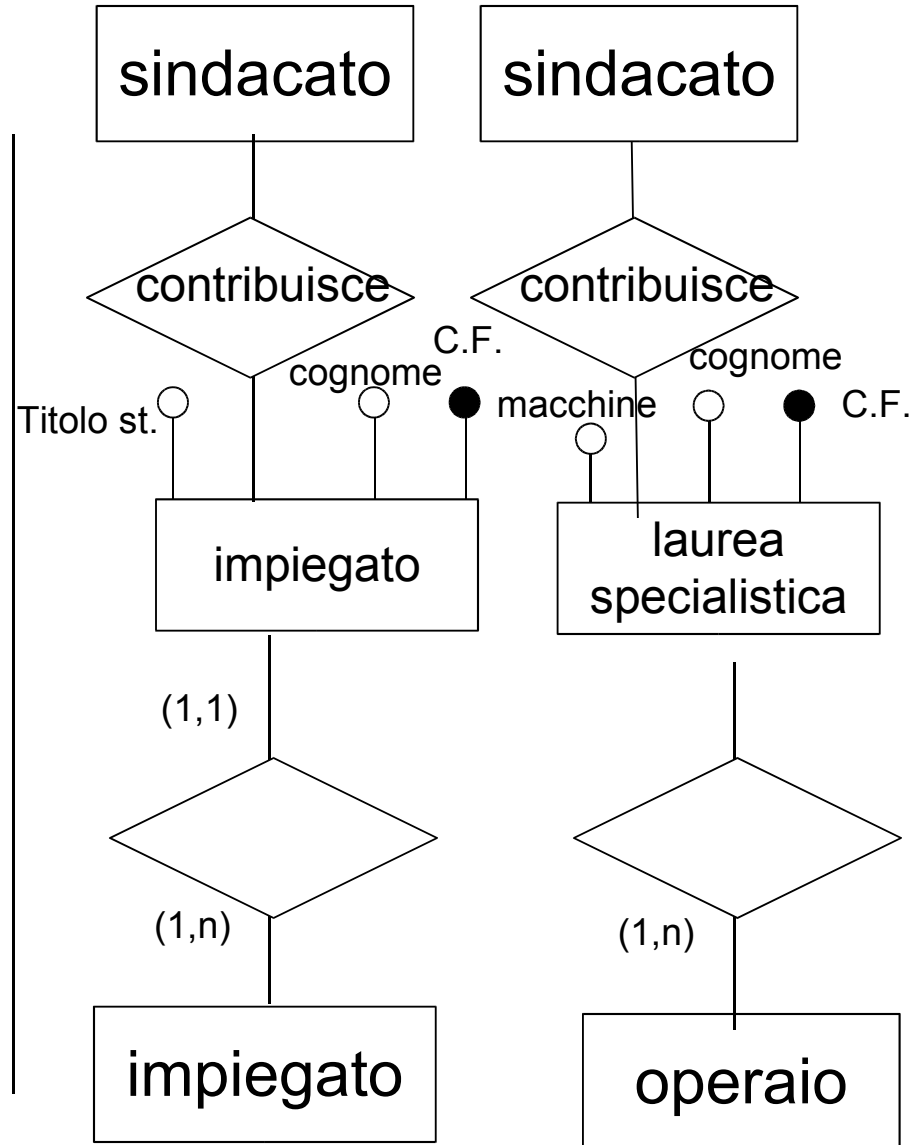
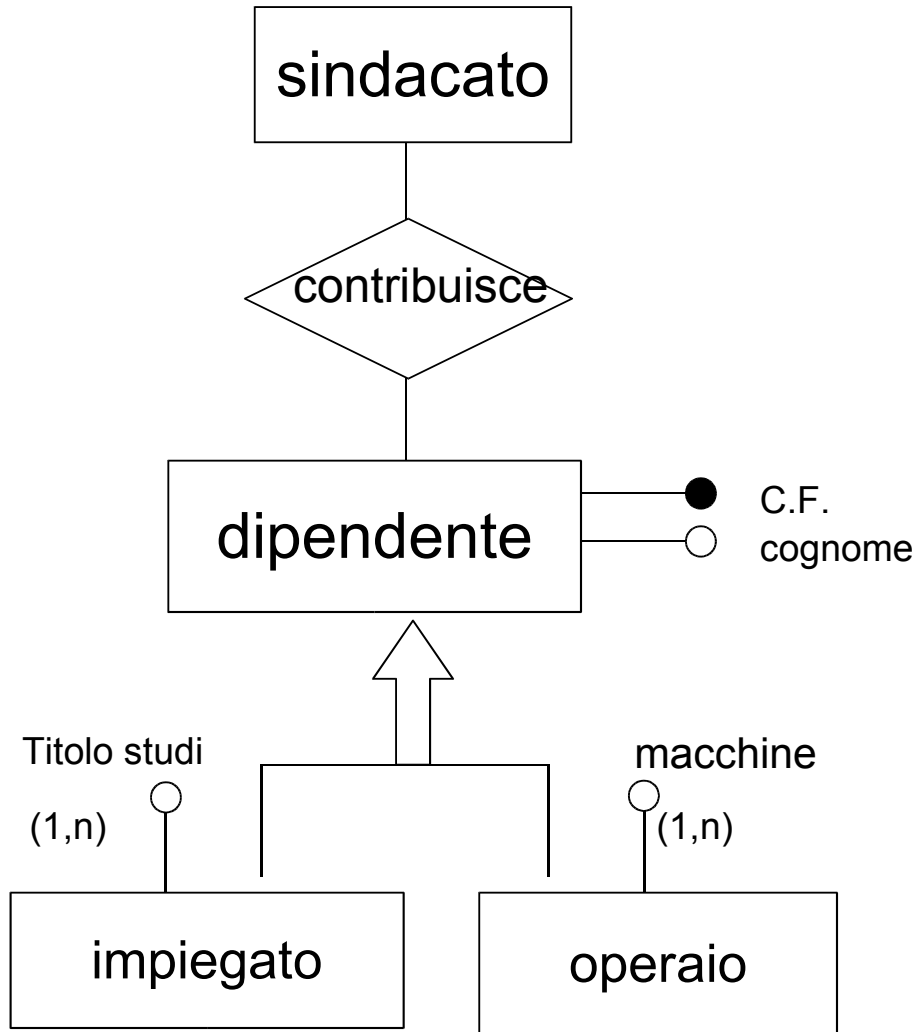
Esempio: collasso verso l'alto



Collasso verso l'alto

- **Dom(Tipo) = {L,D,N}**
- “Tipo” è un attributo selettore che specifica se una singola istanza di Studenti appartiene a una delle N sottoentità
- Copertura
 - **totale esclusiva**: Tipo ha N valori, quante sono le sottoentità
 - **parziale esclusiva**: Tipo ha N+1 valori; il valore in più serve per le istanze che non appartengono a nessuna sottoentità
 - **sovrapposta**: occorrono tanti selettori quante sono le sottoentità, 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 sottoentità si trasportano su E, le eventuali cardinalità minime diventano 0

Esempio: collasso verso il basso



Collasso verso il basso

- Se la **copertura NON è completa** non si può fare
 - non si saprebbe dove mettere le istanze di E che non sono né in E1, né in E2
- Se la copertura **non è esclusiva** introduce **ridondanza**
 - una certa istanza può essere sia in E1 che in E2, e quindi si rappresentano due volte gli attributi che provengono da E

Sostituire con relazione

- È possibile sostituire la gerarchia con una relazione che lega l'entità principale alle singole entità di specializzazione

Cosa conviene fare

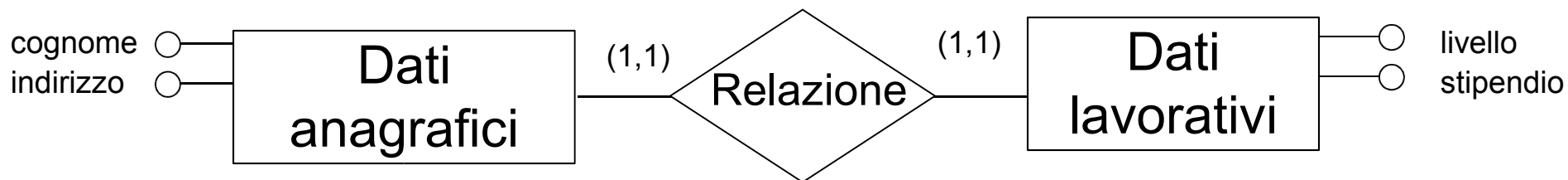
- La scelta fra le alternative si può fare, considerando oltre al numero degli accessi anche l'occupazione di spazio
- È possibile seguire alcune **semplici regole generali** (ovvero: **mantieni insieme ciò che viene usato insieme**)
 - 1. conviene se gli accessi al genitore e alle figlie sono contestuali
 - 2. conviene se gli accessi alle figlie sono distinti (ma 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 gerarchie a più livelli

Partizionamenti e accorpamenti

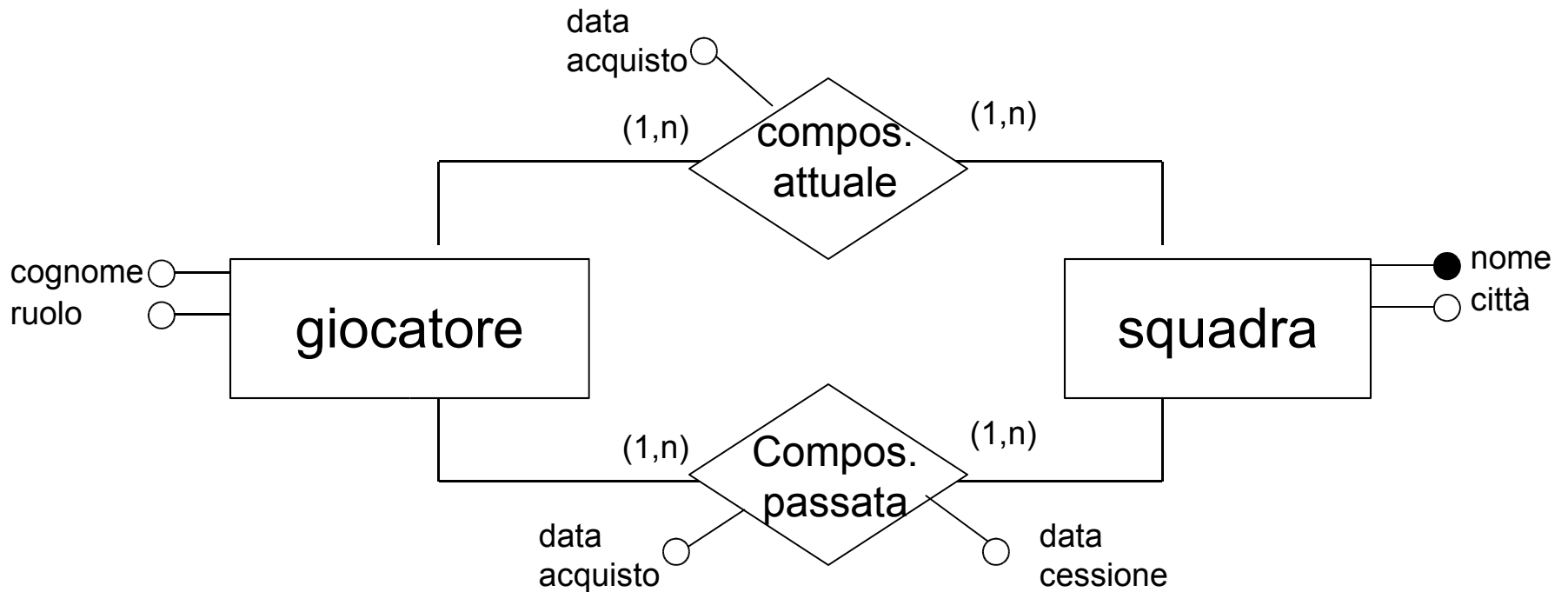
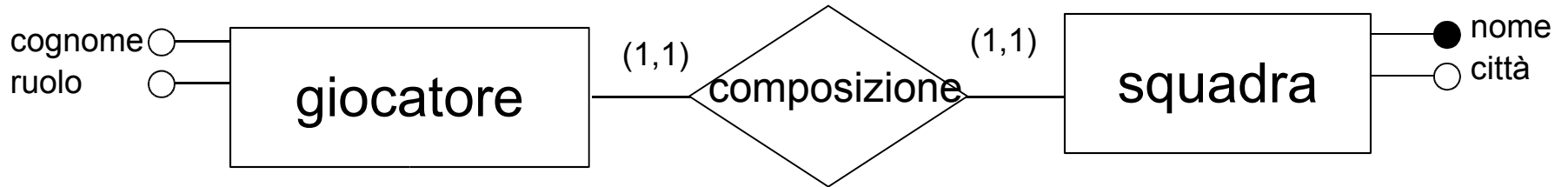
- è possibile ristrutturare lo schema accorpendo o partizionando entità e relazioni
- Tali ristrutturazioni vengono effettuate per rendere più efficienti le operazioni in base al principio già visto, ovvero:
- Gli accessi si riducono:
 - separando attributi di un concetto che vengono acceduti separatamente
 - raggruppando attributi di concetti diversi acceduti insieme
- I casi principali sono:
 - partizionamento “verticale” di entità
 - partizionamento “orizzontale” di relazioni
 - accorpamenti di entità e relazioni
 - eliminazione di attributi multivalore

Partizionamento verticale di entità

Si separano gli attributi in gruppi omogenei



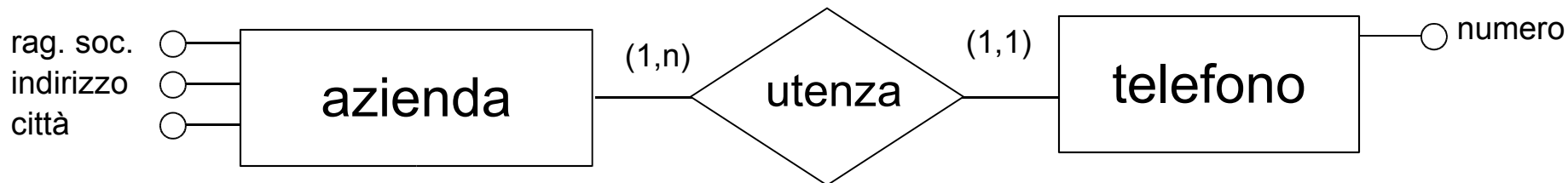
Partizionamento orizzontale di relazioni



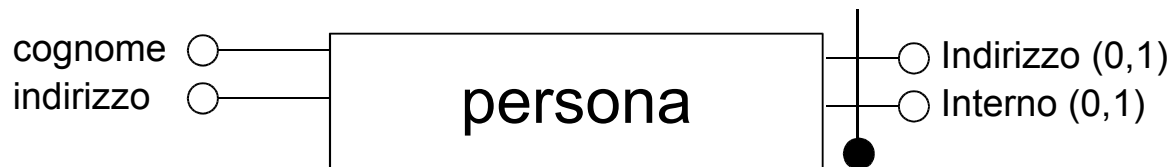
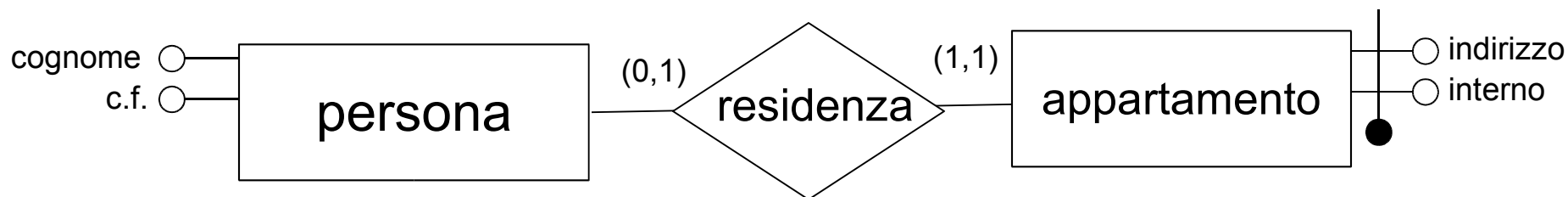
Eliminazione di attributi multivalore

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



Accorpamento di entità



Scelta degli identificatori principali

- È un'operazione indispensabile per la traduzione nel modello relazionale, che 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 dei nuovi attributi (dei “codici”) allo scopo

Traduzione delle entità

- Ogni entità è **tradotta con una tabella** con gli stessi attributi
- La **chiave primaria** coincide con l'**identificatore principale** dell'entità
- Gli **attributi composti** vengono ricorsivamente suddivisi nelle loro componenti, oppure si mappano in un singolo attributo della tabella, il cui dominio va opportunamente definito
- Per brevità, usiamo l'asterisco (*) per indicare la possibilità di **valori nulli**

Traduzione delle entità

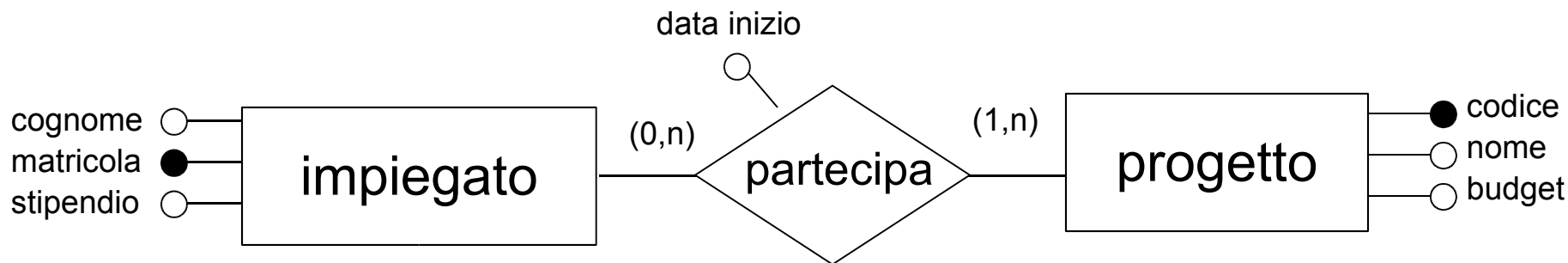


Persona(CF, cognome, nome, via, civico*, città, cap)

Traduzione delle relazioni

- Ogni relazione è **tradotta con una tabella** 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 su vincoli e occupazione inutile di memoria)

Entità e relazione molti a molti



Impiegato(Matricola, Cognome, Stipendio)

Progetto(Codice, Nome, Budget)

Partecipazione(Matricola, Codice, DataInizio)

FK (foreing key): **Matricola** REFERENCES Impiegato

FK (foreing key): **Codice** REFERENCES Progetto

Foreign key

- Non è ovviamente necessario mantenere per gli attributi chiave della tabella che traduce la relazione gli stessi nomi delle chiavi primarie referenziate, ma conviene usare nomi più espressivi
- Ovviamente se le entità collegate hanno un identificatore con lo stesso nome la ridenominazione è obbligatoria!

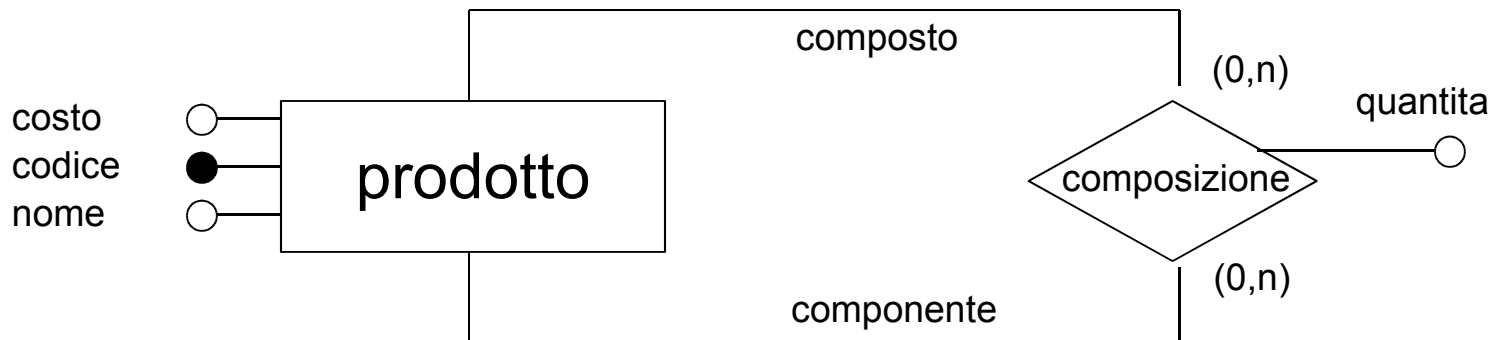
Partecipazione(**Impiegato**, **CodProgetto**, DataInizio)

FK: Impiegato REFERENCES Impiegato

FK: CodProgetto REFERENCES Progetto

Relazioni ad anello molti a molti

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



Prodotto(Codice, Nome, Costo)

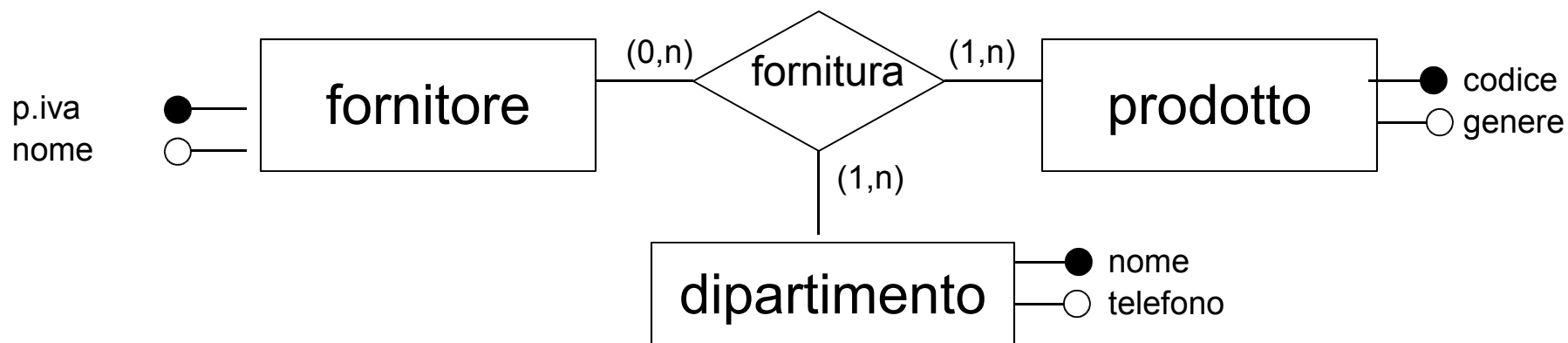
Composizione(**Composto**, **Componente**, Quantità)

FK: Composto REFERENCES Prodotto

FK: Componente REFERENCES Prodotto

Associazioni n-arie molti a molti

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



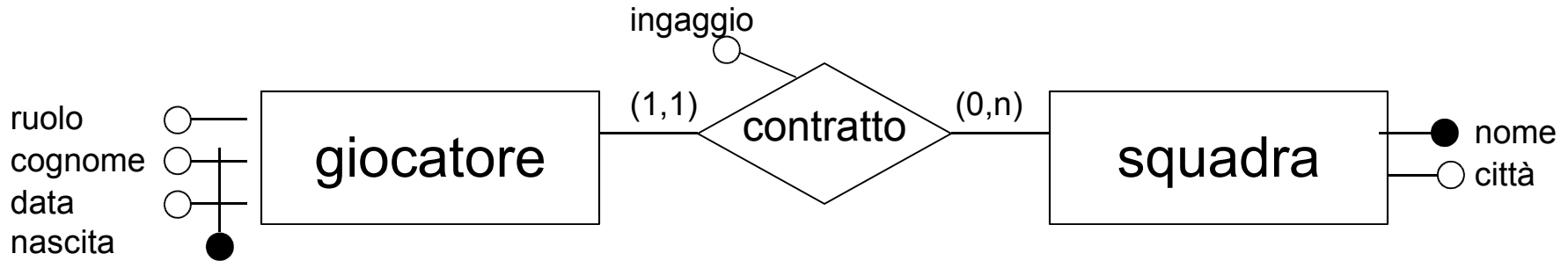
Fornitore(PartitaIVA, Nome)

Prodotto(Codice, Genere)

Dipartimento(Nome, Telefono)

Fornitura(Fornitore, Prodotto, Dipartimento, Quantità)

Relazioni uno a molti



Giocatore(Cognome, DataNascita, Ruolo)

Squadra(Nome, Città)

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
(perché?)

Relazioni uno a molti

- Poiché un giocatore ha un contratto con una sola squadra, nella relazione Contratto un giocatore non può apparire in più tuple
- Si può pertanto pensare anche ad una **soluzione più compatta, facente uso di 2 sole relazioni**

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

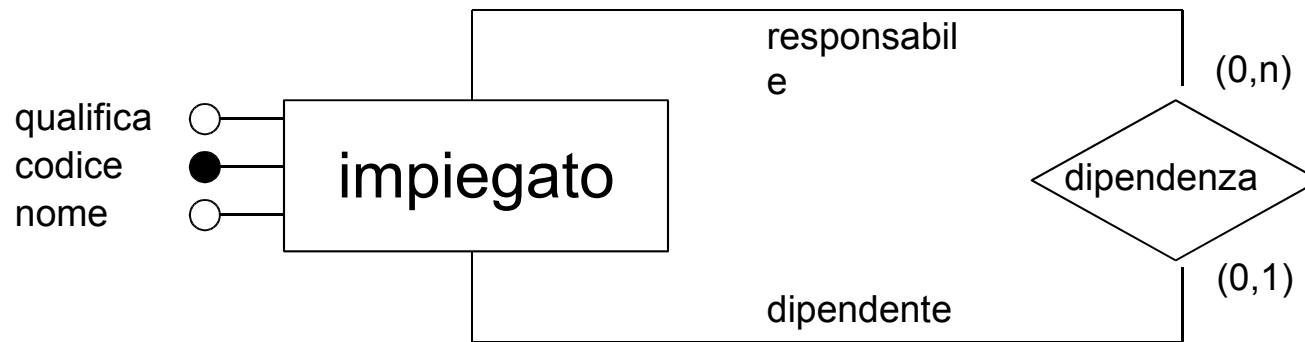
FK: Squadra REFERENCES Squadra

Squadra(Nome, Città)

- che corrisponde a tradurre la relazione insieme a Giocatore (ovvero all'entità che partecipa con cardinalità massima 1)
- Se fosse **min-card(Giocatore,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)

Relazioni ad anello uno a molti

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



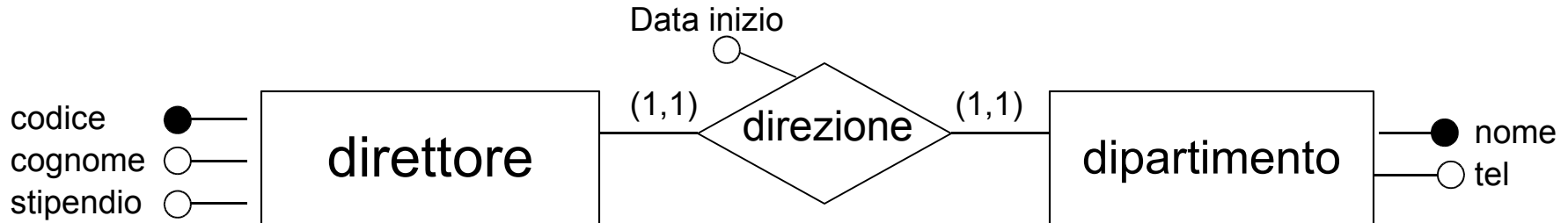
Impiegato(Codice, Nome, Qualifica, Responsabile*)
 FK: Responsabile REFERENCES Impiegato

1 tabella

Impiegato(Codice, Nome, Qualifica)
 Dipendenza(Dipendente, Responsabile)
 FK: Dipendente REFERENCES Impiegato
 FK: Responsabile REFERENCES Impiegato

2 tabelle

Relazioni uno a uno



3 tabelle

Direttore(Codice, Cognome, Stipendio)

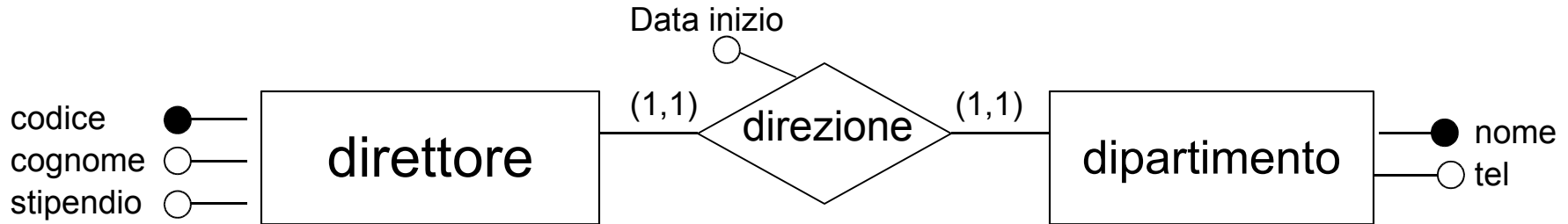
Dipartimento(Nome, Sede, Telefono)

Direzione(Direttore, Dipartimento, DataInizio)

L'identificatore di una delle 2 entità viene scelto come chiave primaria, l'altro dà origine a una chiave alternativa

La scelta dipende dall'importanza relativa delle chiavi

Relazioni uno a uno



2 tabelle

Direttore(Codice, Cognome, Stipendio, Dipartimento, DataInizio)

FK: Dipartimento REFERENCES Dipartimento

Dipartimento(Nome, Sede, Telefono)

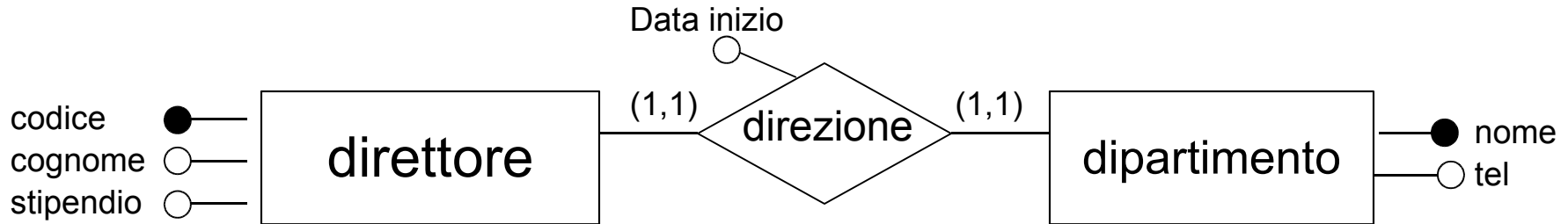
oppure

Direttore(Codice, Cognome, Stipendio)

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

FK: Direttore REFERENCES Direttore

Relazioni uno a uno



2 tabelle

Direttore(Codice, Cognome, Stipendio, Dipartimento, DataInizio)

FK: Dipartimento REFERENCES Dipartimento

Dipartimento(Nome, Sede, Telefono)

oppure

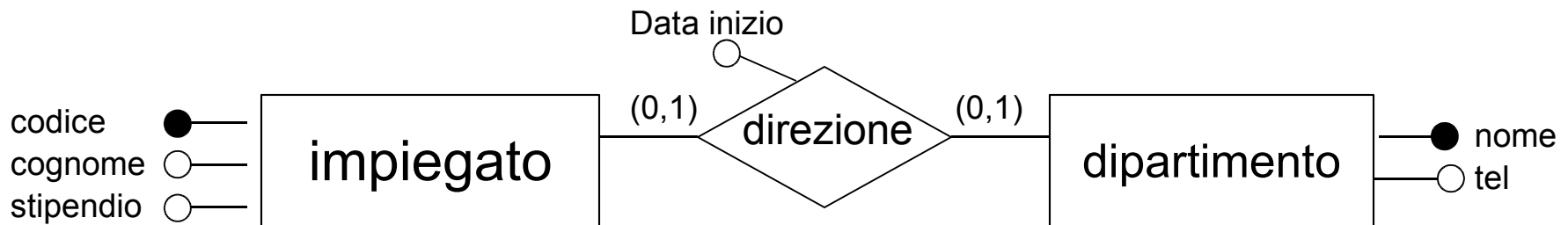
Direttore(Codice, Cognome, Stipendio)

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

FK: Direttore REFERENCES Direttore

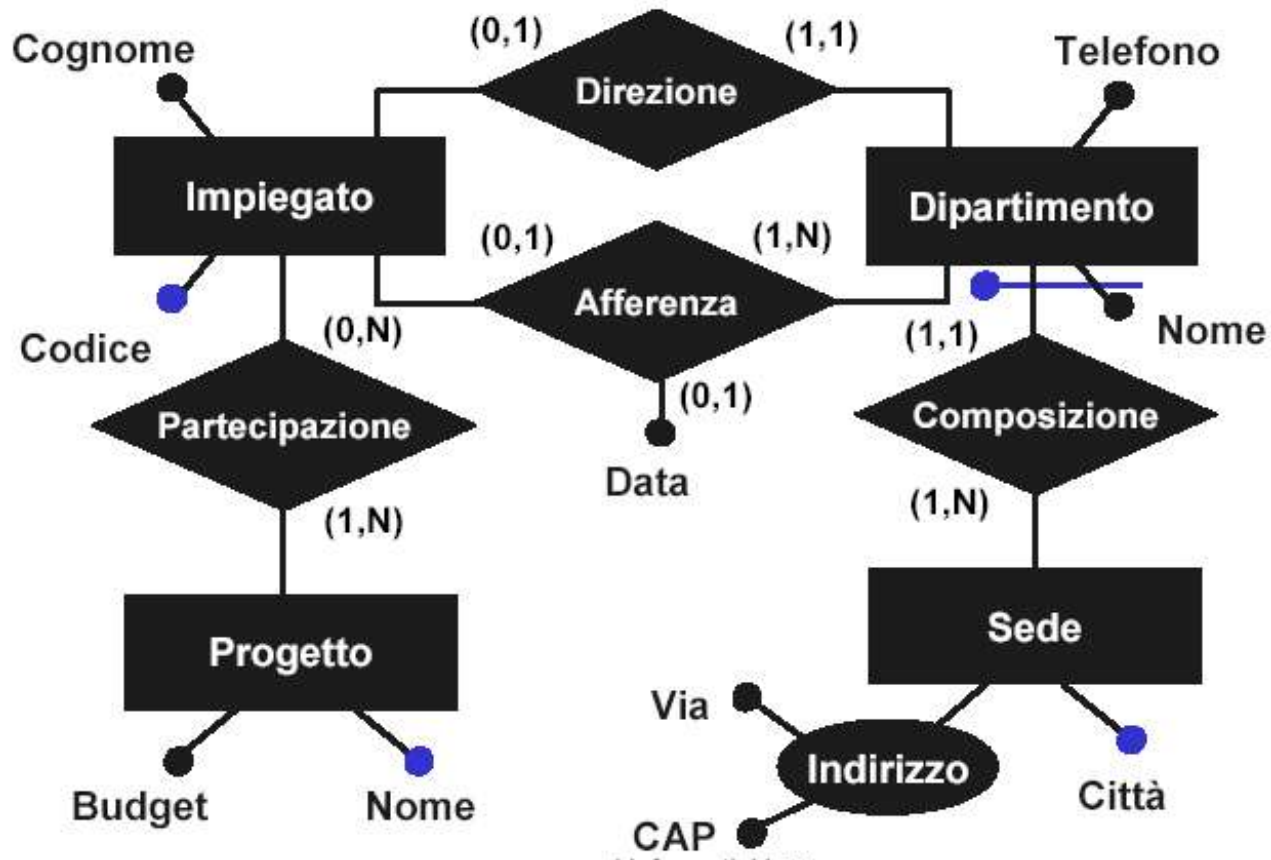
Relazioni ad anello uno a molti

- In linea di principio la traduzione con una sola tabella non andrebbe qui considerata, in quanto corrisponde a un accorpamento di entità, oggetto della fase di ristrutturazione.
- 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 E1 ammetterà valori nulli, e **la chiave primaria si ottiene da E2**
- 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*)`

Esempio di riferimento



Esempio: schema logico relazionale

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