Analisi dei requisiti

- I requisiti devono innanzitutto essere acquisiti
- Le fonti possono essere molto diversificate tra loro:
  - utenti, attraverso:
    - interviste
    - documentazione apposita
  - documentazione esistente:
    - normative (leggi, regolamenti di settore)
    - regolamenti interni, procedure aziendali
    - realizzazioni preesistenti
  - modulistica

- La raccolta dei requisiti è un'attività difficile e non standardizzabile

Esempio: analisi dei requisiti

DB per corsi di formazione

Si vuole realizzare una base di dati per una società che eroga corsi, di cui vogliamo rappresentare i dati dei **partecipanti** ai corsi 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 frequentato (i corsi sono in tutto circa 200) e il giudizio finale.

Rappresentiamo anche i **seminari** che stanno attualmente frequentando e, per ogni giorno, i luoghi e le ore dove sono tenute le lezioni.

Esempio: analisi dei requisiti

DB per corsi di formazione

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.

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

- Raramente i requisisti espressi in linguaggio naturale sono privi di ambiguità. È infatti frequente il caso di
  - **Omonimi**: lo stesso termine viene usato per descrivere concetti differenti (es: libro e copia di libro, posto: di lavoro e geografico)
  - **Sinonimi**: termini diversi vengono usati per descrivere lo stesso concetto (es: studente e partecipante)

- Un modo conveniente per rappresentare sinteticamente i concetti più rilevanti emersi dall'analisi è il glossario dei termini, il cui scopo è fornire per ogni concetto rilevante:
  - Una breve descrizione del concetto
  - Eventuali sinonimi
  - Relazioni con altri concetti del glossario stesso

Glossario dei termini

Termine	Descrizione	Sinonimi	Collegamenti
Partecipante	Persona che partecipa ai corsi. Può essere un dipendente o un professionista	Studente	Corso, Datore
Docente	Docente dei corsi. Può essere un collaboratore esterno	Insegnante	Corso
Corso	Corso organizzato dalla società. Può avere più edizioni	Seminario	Docente, Partecipante
Datore	Datori di lavoro attuali o passati dei partecipanti ai corsi	Posti	Partecipante

Ristrutturazione dei requisiti

- Oltre a costruire il glossario, per semplificare le analisi successive, è utile riformulare i requisiti:
  - Eliminare le omonimie
  - Usare un termine univoco per ogni concetto
  - **Riorganizzare le frasi raggruppandole** in base al concetto cui si riferiscono
  - Nell'esempio:
    - Frasi di carattere generale
    - Frasi riferite ai partecipanti
    - Frasi riferite ai docenti
    - Frasi riferite ai corsi
    - Frasi riferite alle società

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.

Dai concetti allo schema E/R

- Va sempre ricordato che un concetto non è necessariamente sempre un'associazione, un attributo, o altro

  DIPENDE DAL CONTESTO!

- Come regole guida, un concetto verrà rappresentato come
  - Entità
    - se ha proprietà significative e descrive oggetti con esistenza autonoma
  - Attributo
    - se è semplice e non ha proprietà
  - Associazione
    - se correla due o più concetti
  - Generalizzazione/specializzazione
    - se è caso più generale/particolare di un altro

Strategie di progettazione

- Per affrontare progetti complessi è opportuno adottare uno specifico modo di procedere, ovvero una **strategia di progettazione**
- I casi notevoli sono:
  - Strategia **top-down**
    - Si parte da uno schema iniziale molto astratto ma completo, che viene successivamente raffinato fino ad arrivare allo schema finale
  - Strategia **bottom-up**
    - Si suddividono le specifiche in modo da sviluppare semplici schemi parziali ma dettagliati, che poi vengono integrati tra loro
  - Strategia **inside-out**
    - Lo schema si sviluppa “a macchia d’olio”, partendo dai concetti più importanti, aggiungendo quelli ad essi correlati, e così via

Strategie: pro e contro

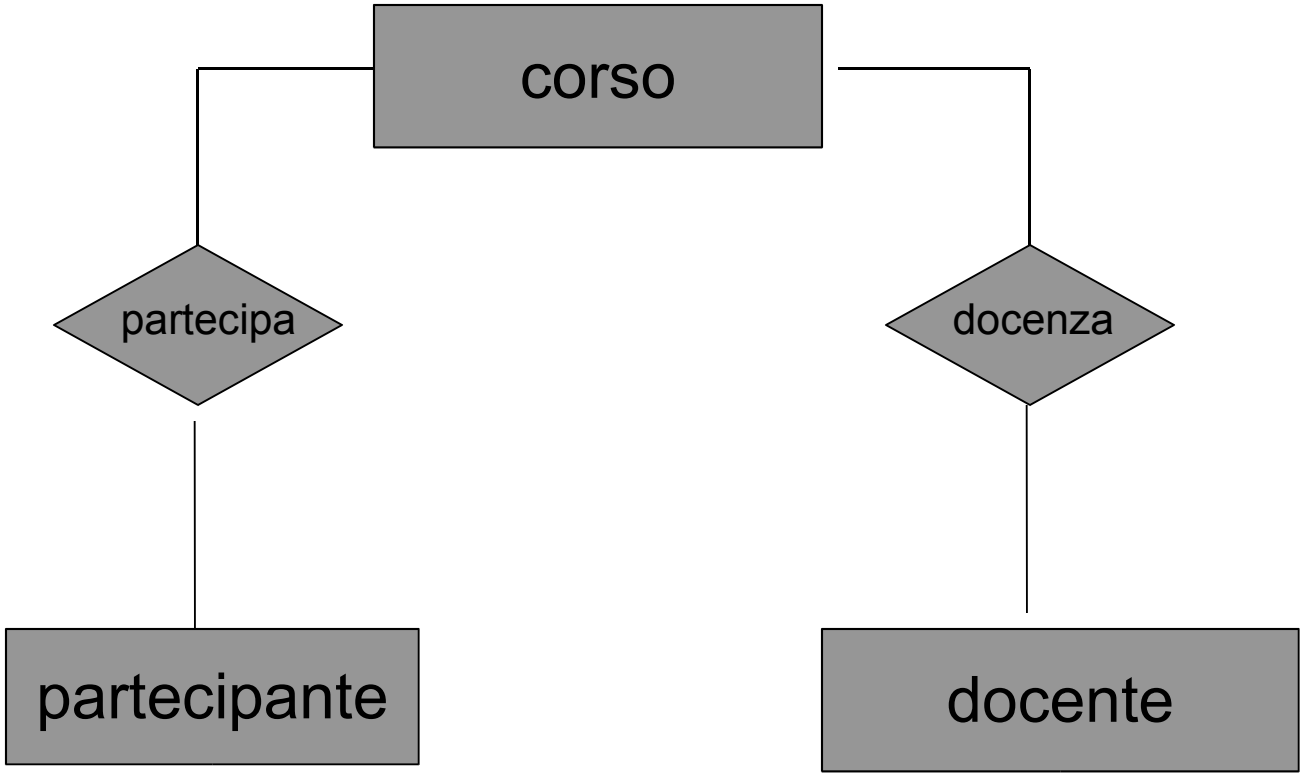
- Top-down
  - **Pro:** non è inizialmente necessario specificare i dettagli
  - **Contro:** richiede sin dall'inizio una **visione globale** del problema, non sempre ottenibile in casi complessi
- Bottom-up
  - **Pro:** permette una ripartizione delle attività
  - **Contro** richiede una **fase di integrazione**
- Inside-out
  - **Pro:** non richiede passi di integrazione
  - **Contro** richiede ad ogni passo di esaminare tutte le specifiche per **trovare i concetti non ancora rappresentati**.

Strategie: approccio misto

- Nella pratica si fa spesso uso di una strategia ibrida, nella quale:
  1 si individuano i concetti principali e si realizza uno **schema scheletro**, che contiene solamente i concetti più importanti
  2 sulla base di questo si può decomporre
  3 poi si raffina, si espande, si integra

- … proseguiamo l'esempio dei corsi di formazione…

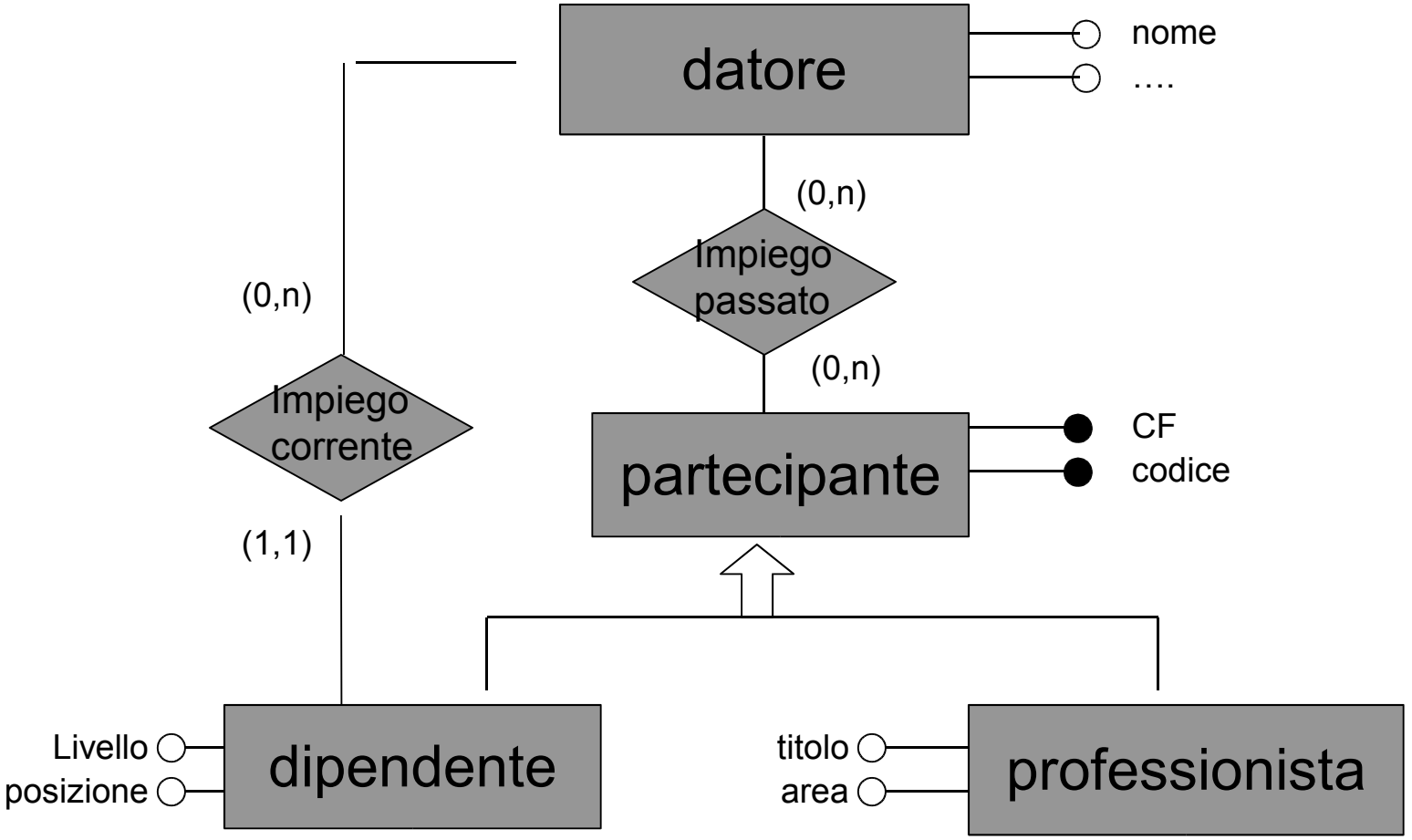
Schema scheletrico



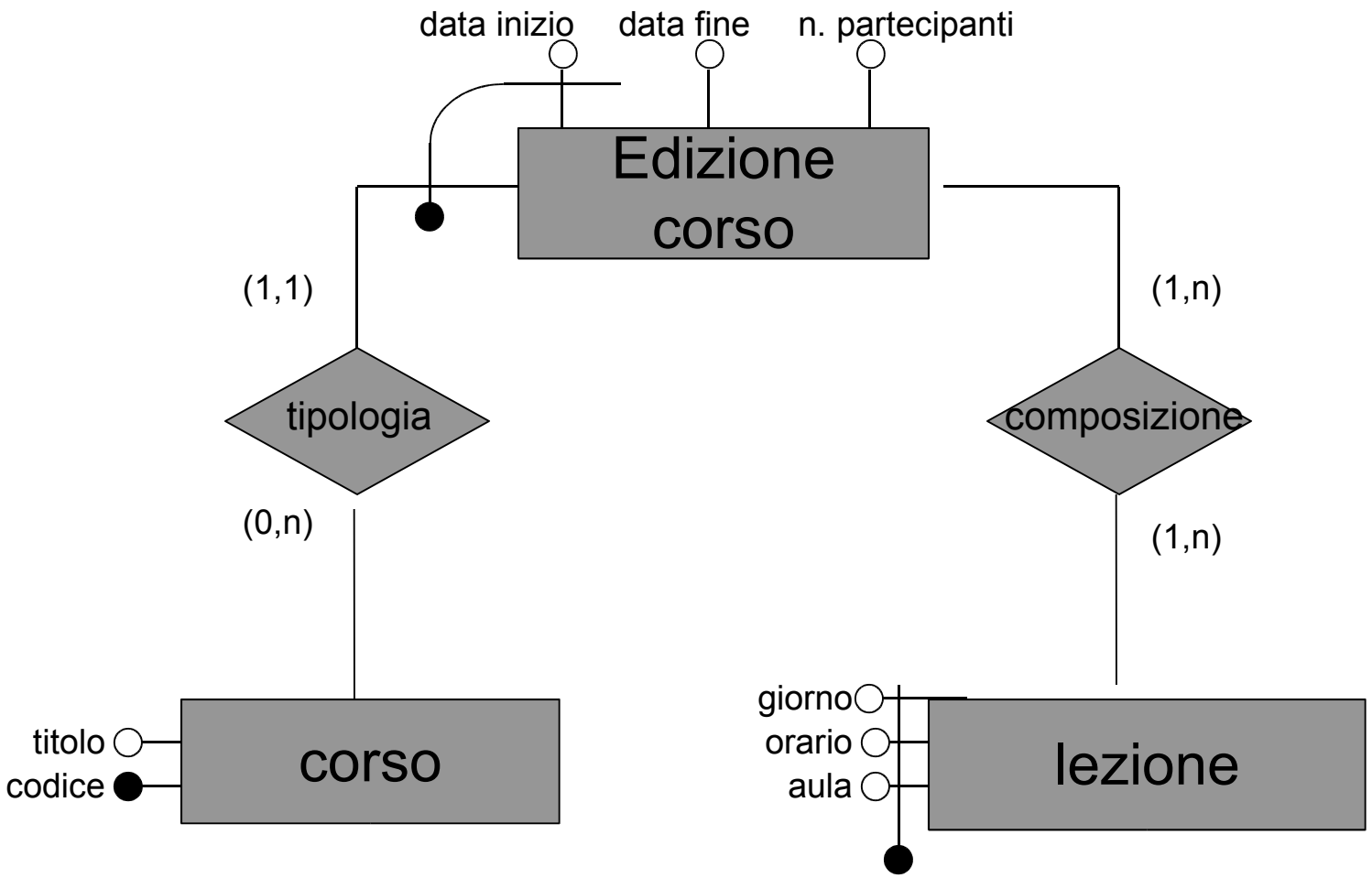
Frasi relative ai partecipanti

- 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.

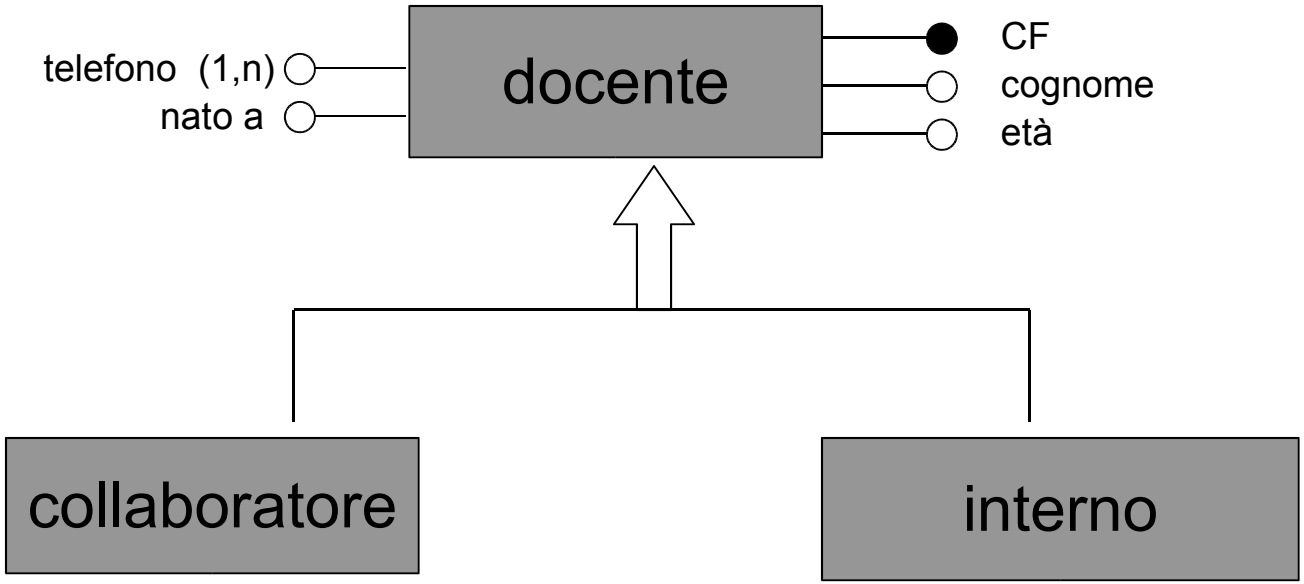
Raffinamento di Partecipante



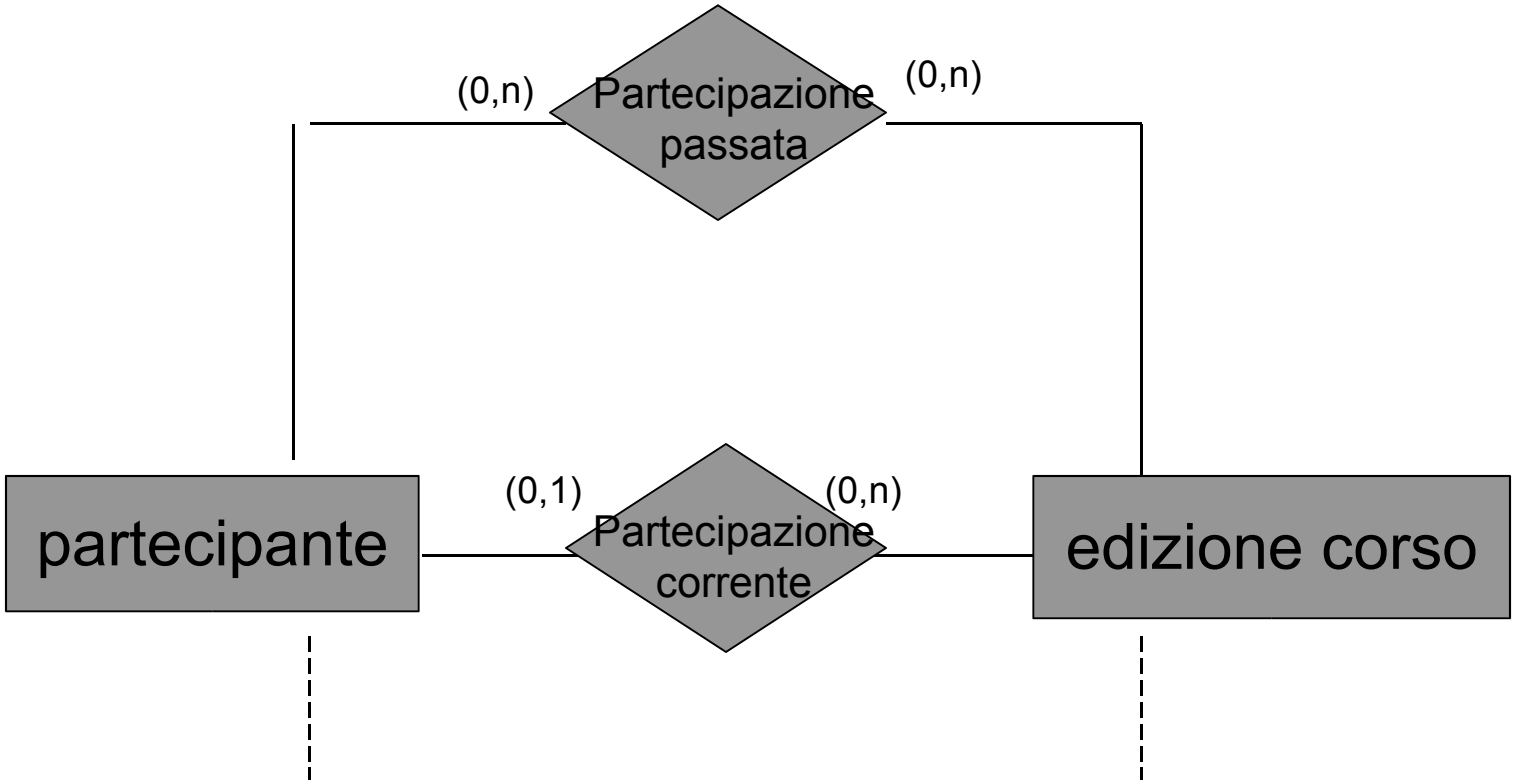
Raffinamento di Corso



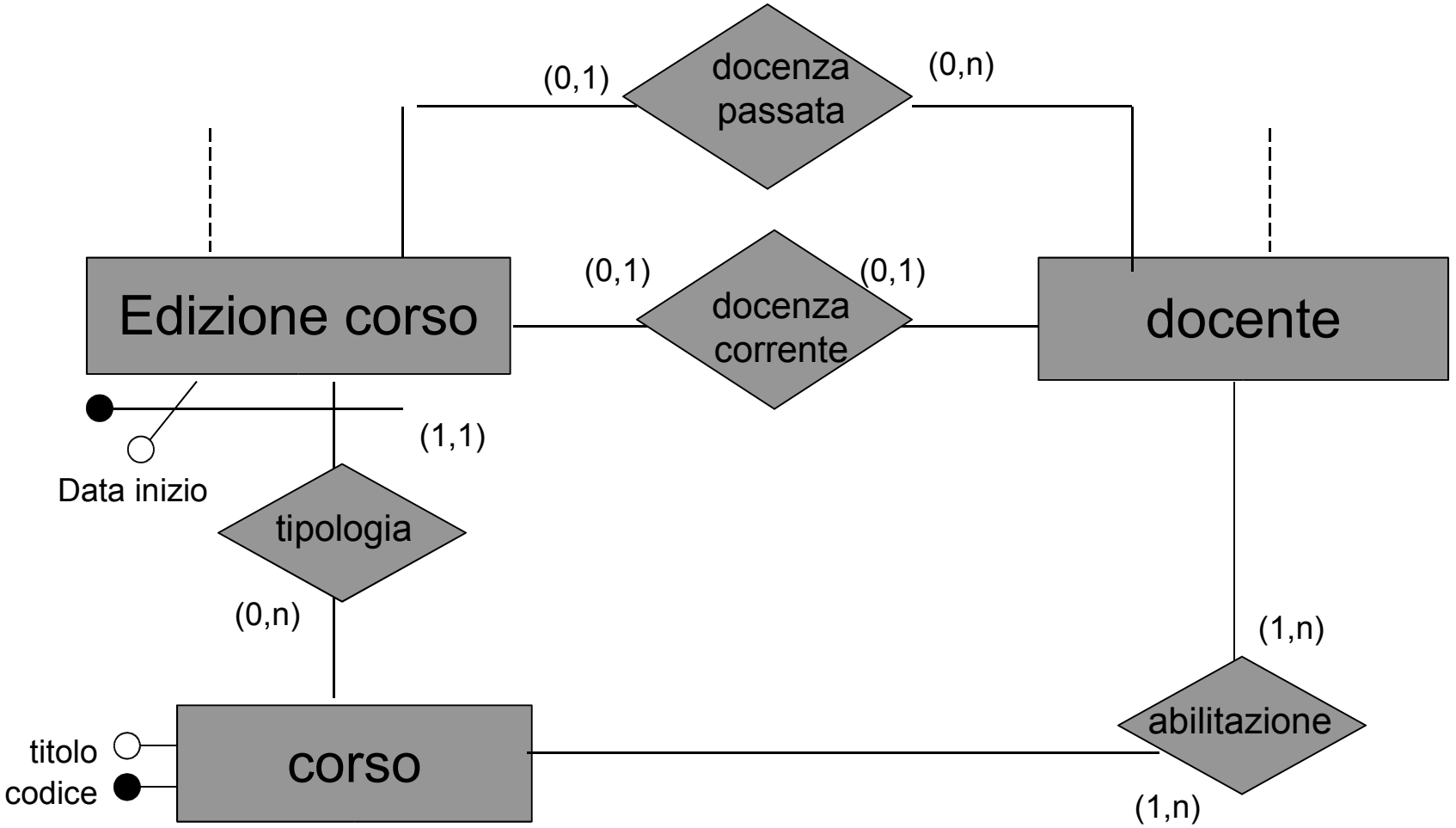
Raffinamento di Docente



Integrazione: partecipante - corso



Integrazione: docente - corso



Qualità di uno schema concettuale

Lo schema E/R deve essere verificato accuratamente per verificare che risponda a requisiti di:

- **Correttezza**
  - Non devono essere presenti errori (sintattici o semantici)
- **Completezza**
  - Tutti i dati di interesse sono specificati
- **Leggibilità**
  - Riguarda anche aspetti prettamente estetici dello schema
- **Minimalità**
  - È importante capire se esistono elementi ridondanti nello schema; in alcuni casi ciò non è un problema, ma può essere viceversa una scelta di progettazione volta a favorire l'esecuzione di certe operazioni

Metodologia della strategia mista

Analisi dei requisiti

- Analizzare i requisiti ed eliminare le ambiguità
- Costruire un glossario dei termini, raggruppare i requisiti

Passo base

- Definire uno schema scheletro con i concetti più rilevanti

Passo di decomposizione (se necessario o appropriato)

- decomporre i requisiti con riferimento ai concetti nello schema scheletro

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

Passo di integrazione (se si è decomposto)

- integrare i vari sottoschemi in uno schema complessivo, facendo riferimento allo schema scheletro

Analisi di qualità (ripetuta e distribuita)

- Verificare le qualità dello schema e modificarlo

Basi di dati

- La progettazione di una base di dati richiede di focalizzare lo sforzo su analisi, progettazione e implementazione della struttura con cui sono organizzati i dati (**modelli di dati**)

- Le funzionalità del sistema non vanno però ignorate

Modelli di dati

- La struttura di una base di dati può essere descritta da modelli, a diversi livelli di astrazione:
- **Modello concettuale**: rappresentazione indipendente da ogni sistema, descrive i concetti del **dominio applicativo**
- **Modello logico**: rappresentazione formale della base di dati indipendente dai dispositivi di archiviazione. Costituisce l'interfaccia tra il DBMS e gli utenti (o le applicazioni)
- **Modello fisico**: determina come il DBMS archivia i dati

Modelli concettuali

- Esistono diversi modelli di database:
  - Gerarchico
  - Reticolare
  - **Relazionale**
  - Ad oggetti

Modello relazionale

- Il modello **relazionale** è quello a più larga diffusione: praticamente tutti i database comunemente utilizzati sono relazionali

- Il modello ad oggetti è il più recente, ma ancora poco utilizzato

Modelli concettuali E-R

- Lo schema **Entity-Relationship** è uno degli strumenti più utilizzati per la modellazione concettuale del dominio applicativo
- Il modello E-R prevede due concetti di base
  - **entità** che rappresentano elementi autonomi del dominio
  - **relazioni** che descrivono correlazioni logiche tra le entità
- Si tratta di un modello semiformale, realizzato tramite diagrammi (eventualmente accompagnati da documenti di specifica)

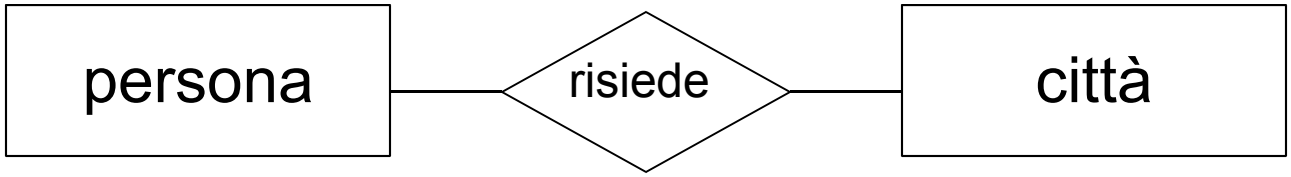
Entità – Relazione: entità

- L'entità è un insieme (classe) di **oggetti** della realtà di interesse **che possiedono caratteristiche comuni** (es. persone, automobili, …) e che hanno esistenza "autonoma"
- L'istanza (elemento) di un'entità è uno specifico oggetto appartenente a quella entità (es. io, la mia auto, …)

- Graficamente un'entità si rappresenta con un rettangolo

Entità – Relazione: relazione

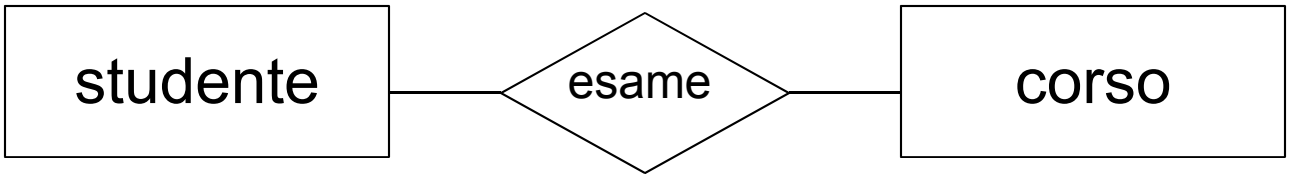
- La relazione rappresenta un legame logico tra entità, rilevante nella realtà che si sta considerando
- Istanza di associazione: combinazione (aggregazione) di istanze delle entità che prendono parte all'associazione
- Graficamente un'associazione si rappresenta con un rombo:



Se p è un'istanza di Persona e c è un'istanza di Città, la coppia (p, c) è un'istanza dell'associazione Risiede

Istanze di relazioni

- Per definizione l'insieme delle istanze di un'associazione è un **sottoinsieme del prodotto Cartesiano degli insiemi delle istanze di entità che partecipano all'associazione**
- Ne segue che non ci possono essere istanze ripetute nell'associazione

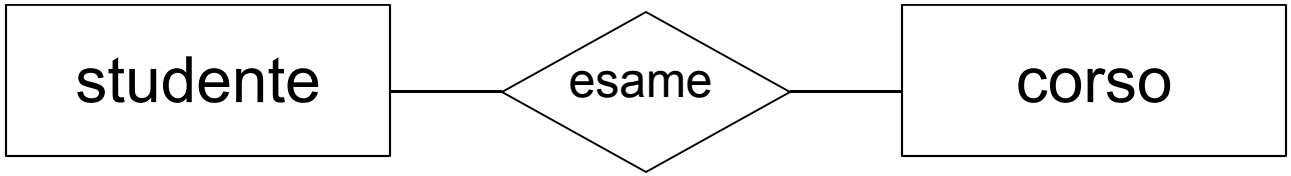


- Se s è uno Studente e c un Corso, la coppia (s,c) può comparire un'unica volta nell'insieme delle istanze di Esame
  - vedremo più avanti come si può rappresentare la possibilità di sostenere più volte lo stesso esame

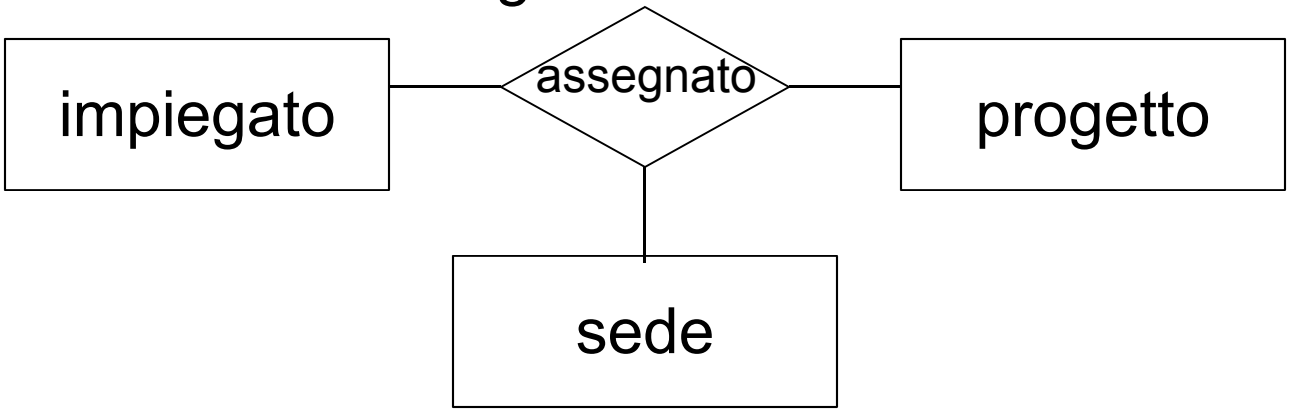
Grado delle relazioni

- È il numero di istanze di entità che sono coinvolte in un'istanza della relazione

relazione binaria: grado = 2

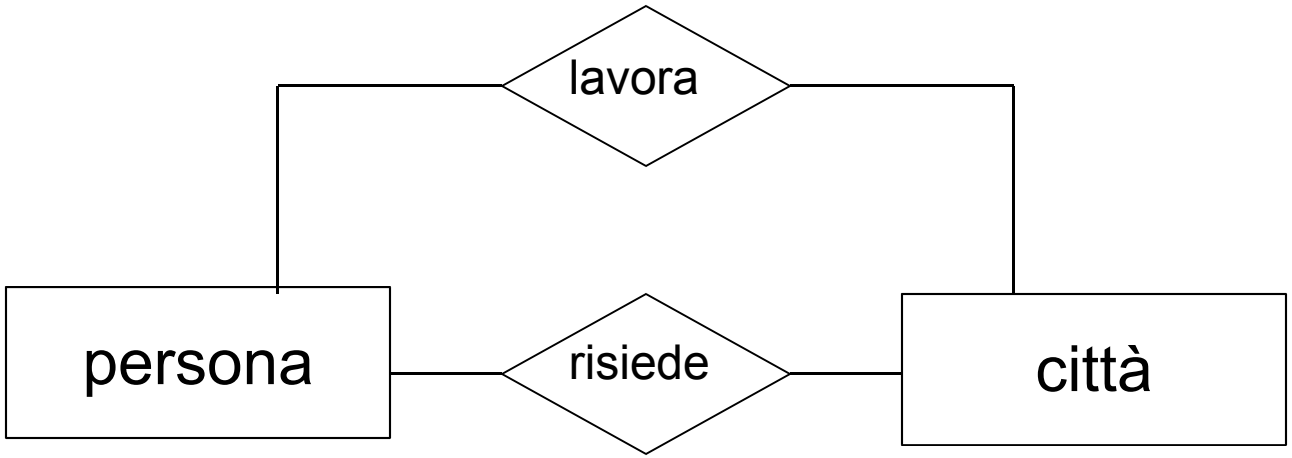


relazione ternaria: grado = 3



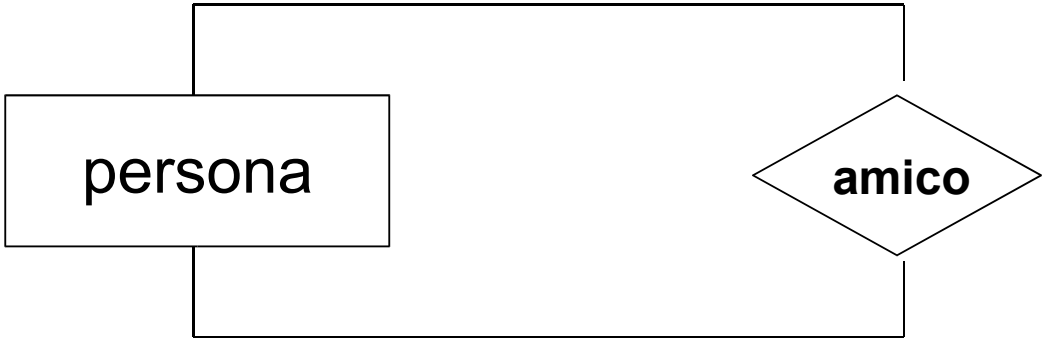
Più relazioni tra le stesse entità

- È possibile stabilire più relazioni, di diverso significato, tra le stesse entità



Relazioni ad anello

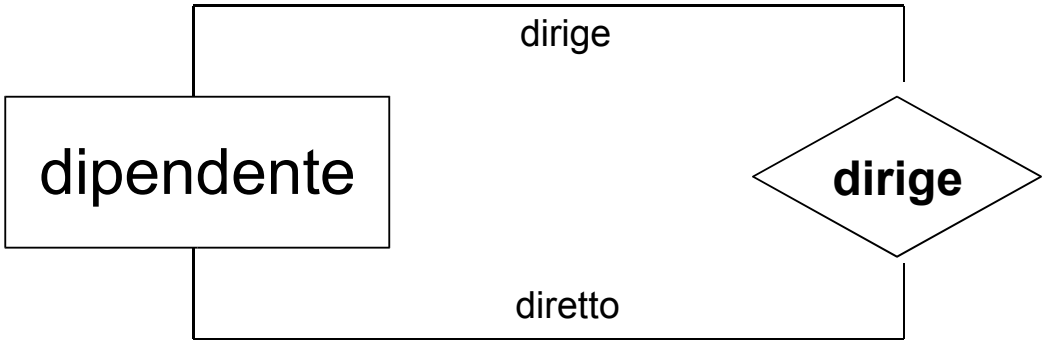
- Una relazione ad anello coinvolge più volte la stessa entità, e quindi mette in relazione tra loro le istanze di una stessa entità



- Una relazione ad anello può essere:
  - **Simmetrica** $(a,b) \in A \Rightarrow (b,a) \in A$
  - **Riflessiva** $(a,a) \in A$
  - **Transitiva** $(a,b) \in A , (b,c) \in A \Rightarrow (a,c) \in A$
- La relazione amico è simmetrica ma né riflessiva né transitiva

Relazioni ad anello

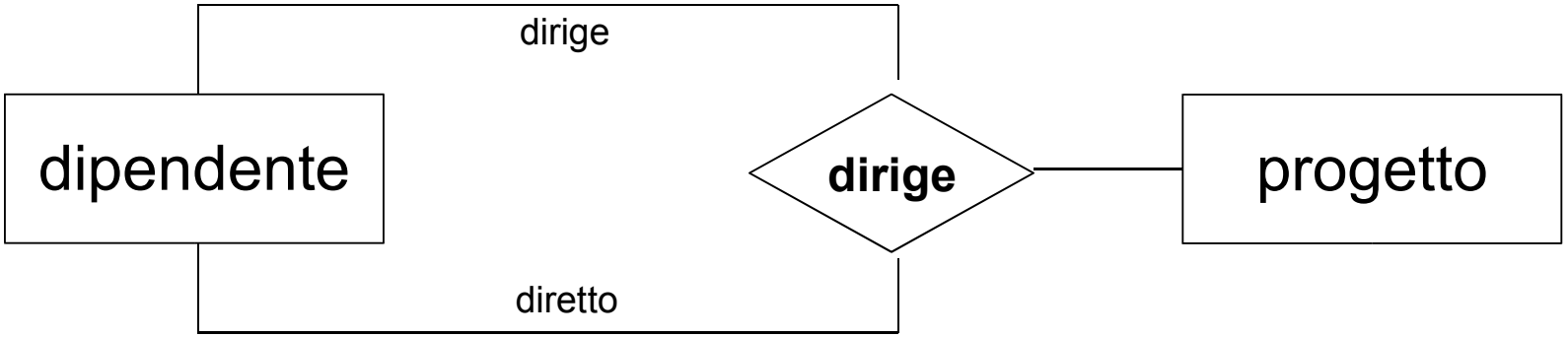
- Nelle relazioni ad anello **non simmetriche** è necessario specificare, per ogni ramo dell'associazione, il relativo ruolo



- L'importanza dei ruoli diventerà evidente appena introdurremo i vincoli di cardinalità

Relazioni ad anello

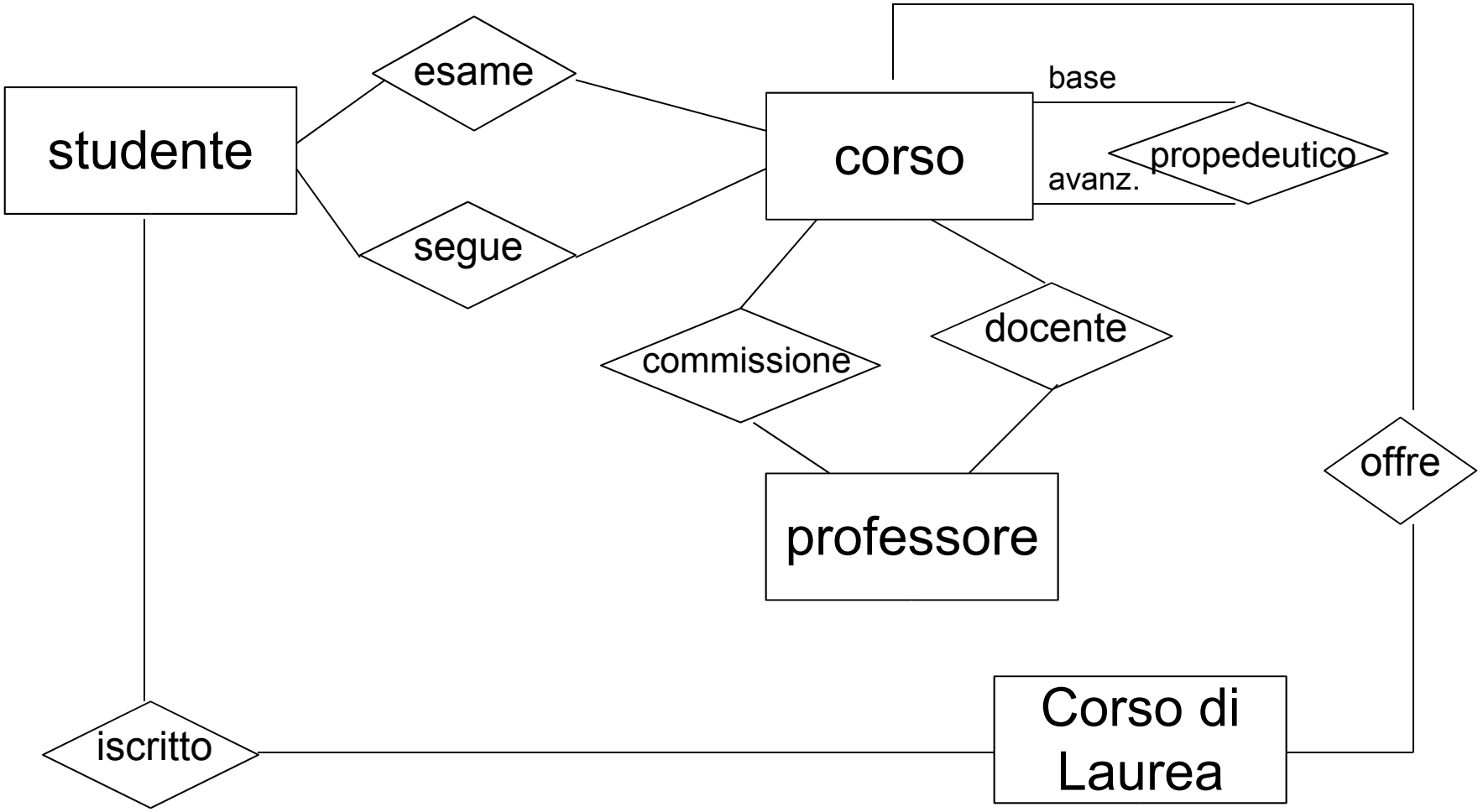
- È possibile avere anelli anche in relazioni n-arie generiche (n > 2)



- Il significato dell'istanza (d1,d2,p) è:

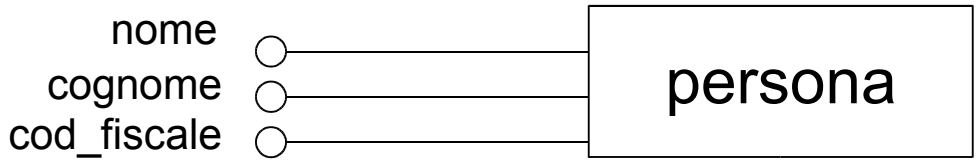
  il dipendente d1 *dirige il dipendente* d2 *all'interno del progetto* p

Uno schema E-R (incompleto)



Attributi

- Un attributo è una proprietà elementare di un'entità o di una relazione
- Graficamente:

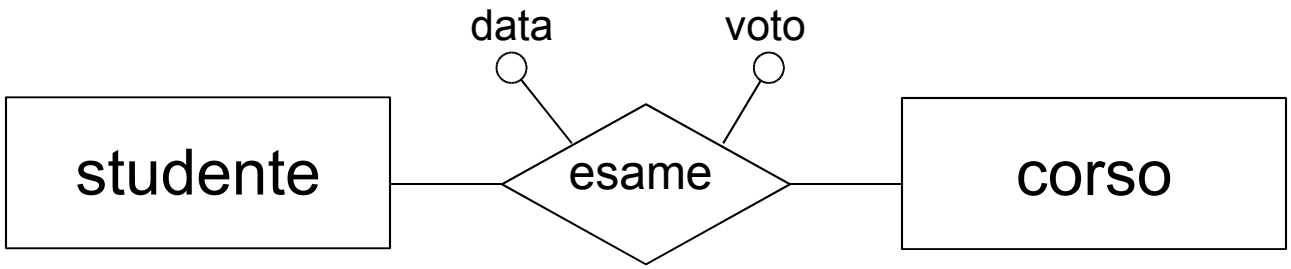


nome, cognome, cod_fiscale sono tutti attributi di Persona

- Ogni attributo è definito su un dominio di valori
- Quindi un attributo associa ad ogni istanza di entità o associazione un valore del corrispondente dominio

Attributi: di entità o di relazione?

- È importante fare attenzione a dove si specificano gli attributi!

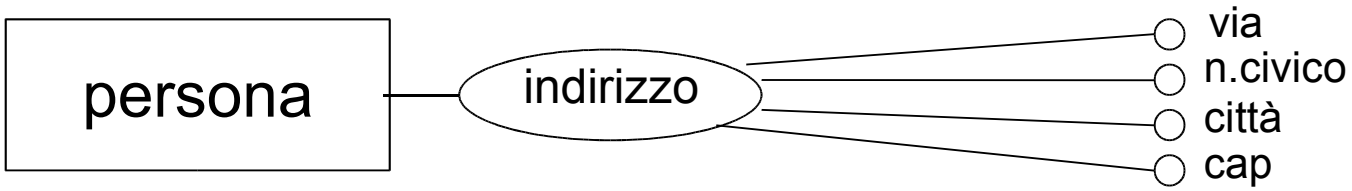


  - data e voto non sono proprietà né di uno Studente né di un Corso, ma del legame Studente-Corso che si crea in occasione di un esame

Attributi composti

- Sono attributi che si ottengono **aggregando** altri (sotto-)attributi, i quali presentano una forte affinità nel loro uso e significato

  es: via, n. civico, città e cap formano l'attributo composto indirizzo



- Si noti che se A è composto dagli attributi A1, A2, …, An con rispettivi domini D1, D2, …, Dn, allora il dominio di A è il prodotto Cartesiano D = D1 x D2 x … x Dn

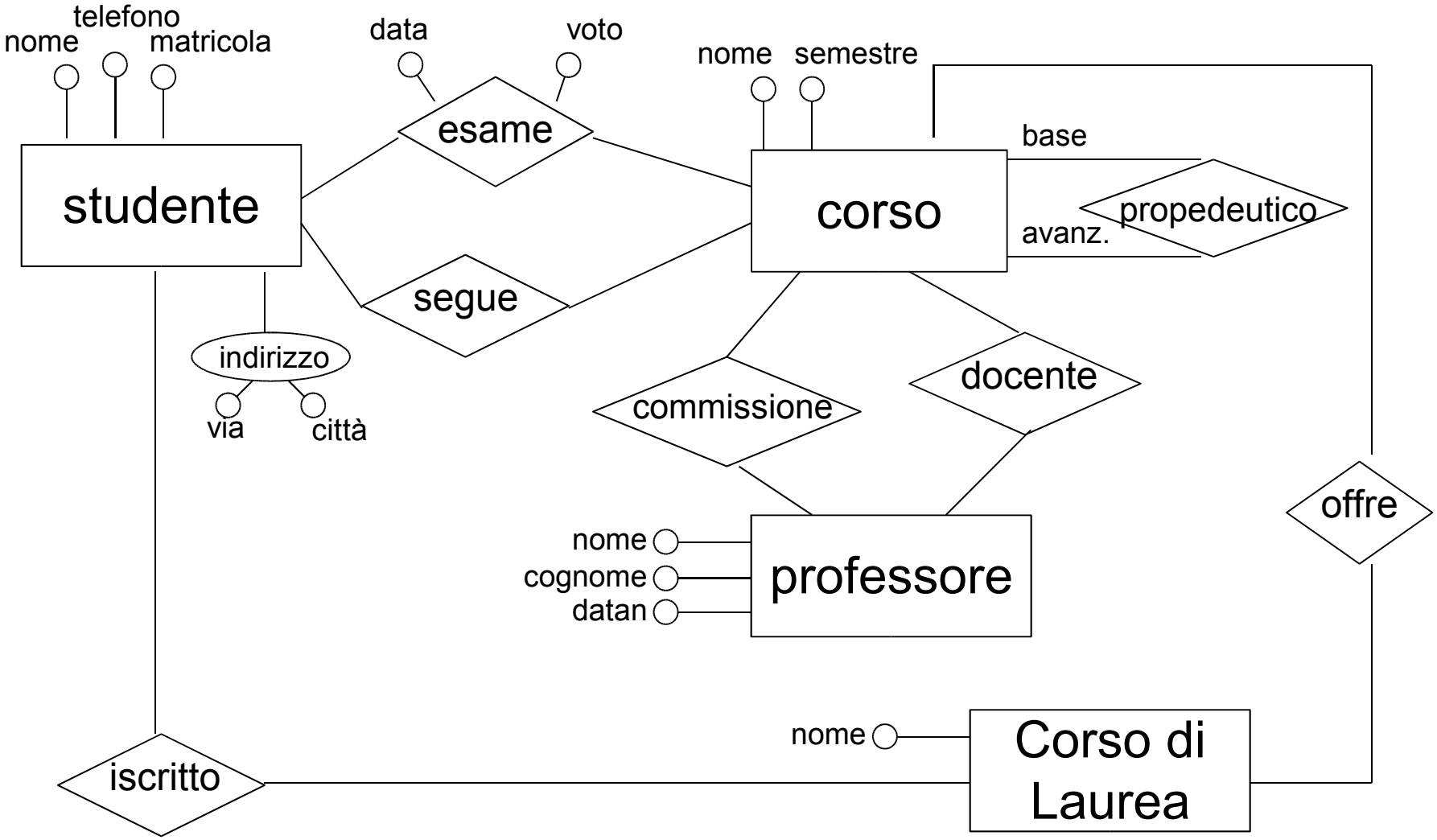
- Un attributo **non composto** viene anche detto **semplice**

Esempio di attributi e domini

- Per l'entità Persona, gli attributi e i relativi domini potrebbero essere:
  - nome: stringa(20)
  - cognome: stringa(20)
  - cod_fiscale: stringa(16)
  - data_di_nascita: giorno x mese x anno
  - titolo_di_studio: stringa(50)

  dove i domini giorno, mese, ed anno sono:
  - giorno = 1, ..., 31
  - mese = {Gen,Feb,Mar,Apr,Mag,Giu,Lug,Ago,Set,Ott,Nov,Dic}
  - anno = 1900, ..., 2100

Uno schema E-R (ancora incompleto)



Vincoli

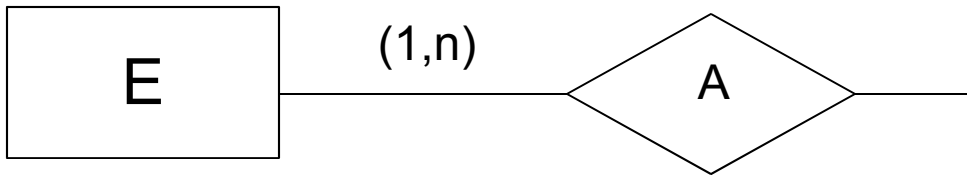
- In ogni schema E/R sono presenti dei vincoli
- Alcuni sono **impliciti**, in quanto dipendono dalla semantica stessa dei costrutti del modello:
  - **ogni istanza di relazione deve riferirsi ad istanze di entità**
  - istanze diverse della stessa relazione devono riferirsi a differenti combinazioni di istanze delle entità partecipanti all'associazione
  - … ed altri che vedremo

- Altri vincoli sono **espliciti**, e vengono definiti da chi progetta lo schema E/R sulla base della conoscenza della realtà che si sta modellando
  - vincoli di **cardinalità** (per relazioni e attributi)
  - vincoli di **identificazione**

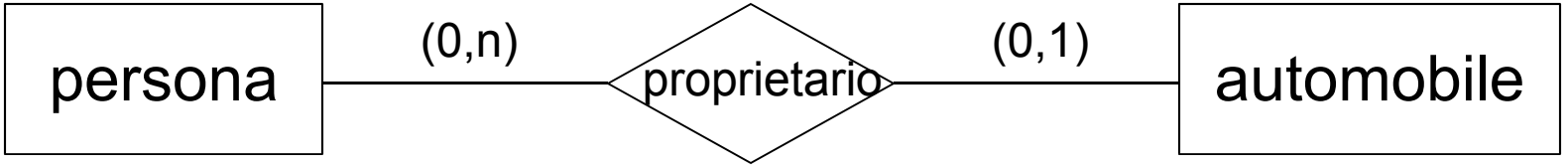
Relazione: vincoli di cardinalità

- Sono coppie di valori **(min-card,max-card) associati a ogni entità** che partecipa a un'associazione, che specificano il **numero minimo e massimo di istanze della relazione a cui un'istanza dell'entità può partecipare**
- Vale quanto già visto, abbiamo solo un modo più compatto per rappresentare i vincoli
- Ad esempio, se i vincoli di cardinalità per un'entità E relativamente a un'associazione A sono (1,n) questo significa:
  - ogni istanza di E partecipa almeno ad una istanza di A
  - ogni istanza di E può partecipare a più istanze di A

- Graficamente:



Vincoli di cardinalità: esempio



- min-card(Automobile,Proprietario) = 0: esistono automobili non possedute da alcuna persona
- min-card(Persona,Proprietario) = 0: esistono persone che non posseggono alcuna automobile
- max-card(Persona,Proprietario) = n: ogni persona può essere proprietaria di un numero arbitrario di automobili
- max-card(Automobile,Proprietario) = 1: ogni automobile può avere al più un proprietario

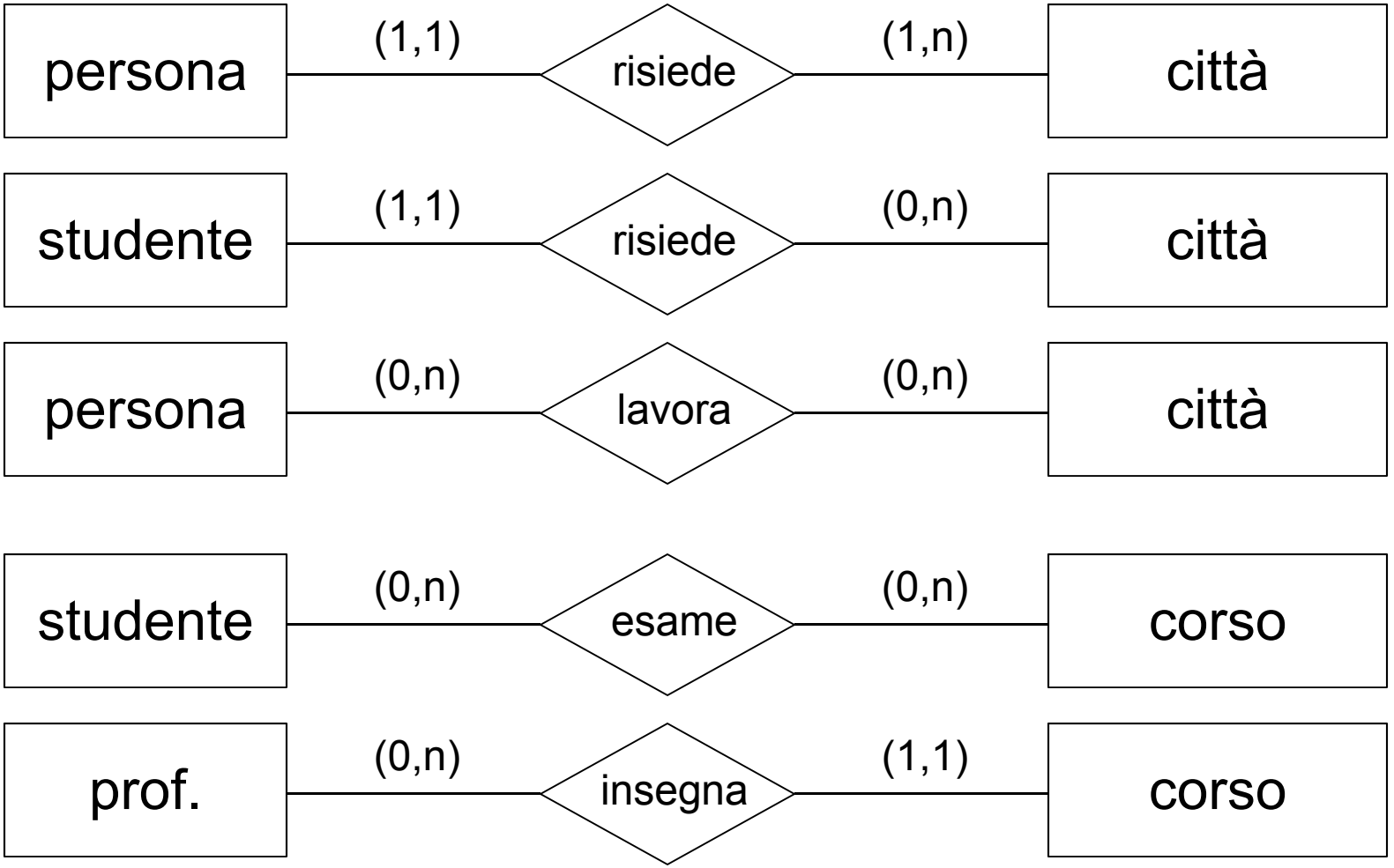
Si noti che i vincoli si possono stabilire correttamente solo se è ben chiaro cosa rappresentano le diverse entità!

Tipi di relazione: terminologia

- Nel caso di una relazione binaria A tra due entità E1 ed E2 (non necessariamente distinte), si dice che:
- A è **uno a uno** se le cardinalità massime di entrambe le entità rispetto ad A sono 1
- A è **uno a molti** se max -card(E1,A) = 1 e max-card(E2,A) = n, o viceversa
- A è **molti a molti** se max-card(E1,A) = n e max-card(E2,A) = n

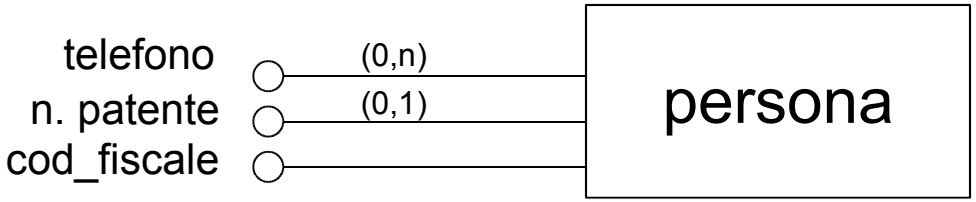
- Si dice inoltre che:
  - La partecipazione di E1 in A è **opzionale** se **min-card(E1,A) = 0**
  - La partecipazione di E1 in A è **obbligatoria** (o **totale**) se **min-card(E1,A) = 1**

Vincoli di cardinalità: esempi

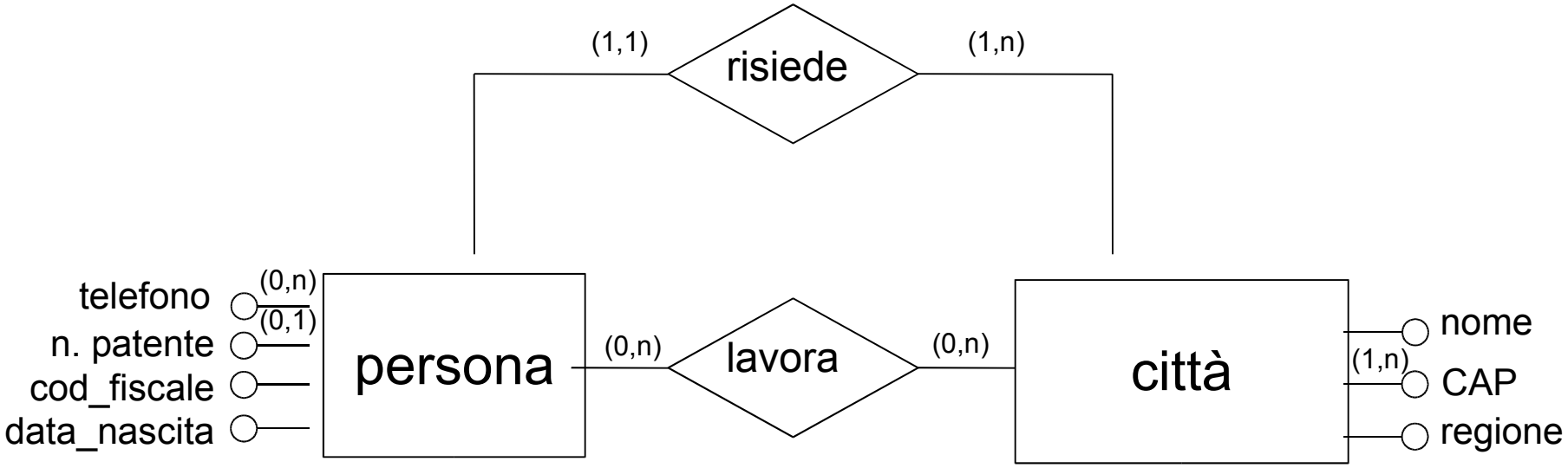


Attributi: vincoli di cardinalità

- Anche per gli attributi è possibile specificare il numero minimo e massimo di valori dell'attributo che possono essere associati ad un'istanza della corrispondente relazione o entità
- Graficamente si può indicare la coppia **(min-card,max-card)** sulla linea che congiunge l'attributo alla relazione/entità, o affianco al nome dell'attributo
  - se non si indica niente il valore di default è (1,1)
- Si parla di attributi:
  - **opzionali**: se la cardinalità minima è 0 (es. n. patente)
  - **monovalore**: se la cardinalità massima è 1 (es. cod_fiscale)
  - **multivalore** (o **ripetuti**): se la cardinalità massima è n (es. telefono)



Esempio con vincoli di cardinalità



Attributi composti

- Nel caso di presenza di più attributi multivalore, la creazione di un attributo composto può rendersi necessaria per evitare ambiguità

- Ad esempio, se una persona ha più indirizzi…

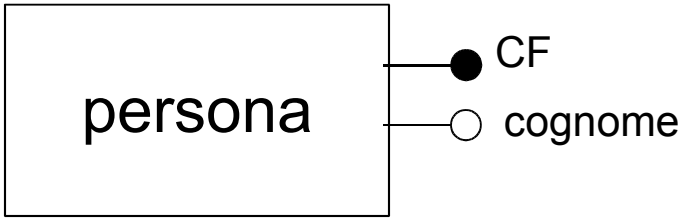
Identificatori

- Un identificatore ha lo scopo di permettere **l'individuazione univoca delle istanze di un'entità**
- Deve valere anche la **minimalità**: nessun sottoinsieme proprio dell'identificatore deve a sua volta essere un identificatore
- Per definire un identificatore per un'entità E si hanno due possibilità di base:
  - **Identificatore interno**: si usano uno o più attributi di E
  - **Identificatore esterno**: si usano altre (una o più) entità, collegate a E da relazioni, più eventuali attributi di E
- Talvolta quando l'identificatore usa sia altre entità che attributi propri si parla di identificatore misto

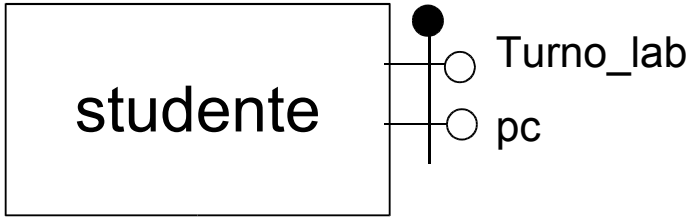
- Se il numero di elementi (attributi o entità) che costituiscono l'identificatore è pari a 1 si parla di **identificatore semplice**, altrimenti l'identificatore è **composto**

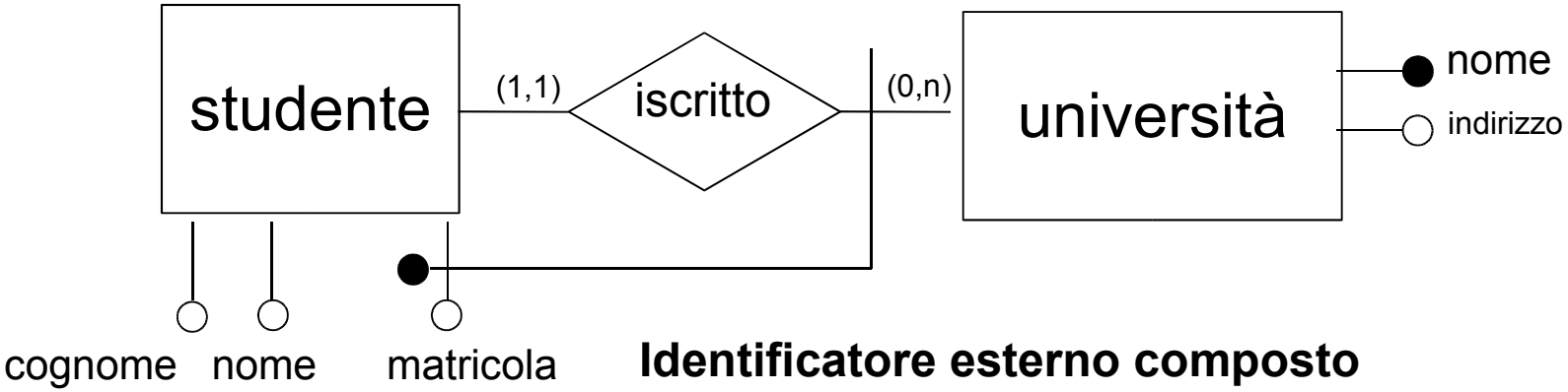
Identificatori interni ed esterni

Identificatore interno semplice



Identificatore interno composto





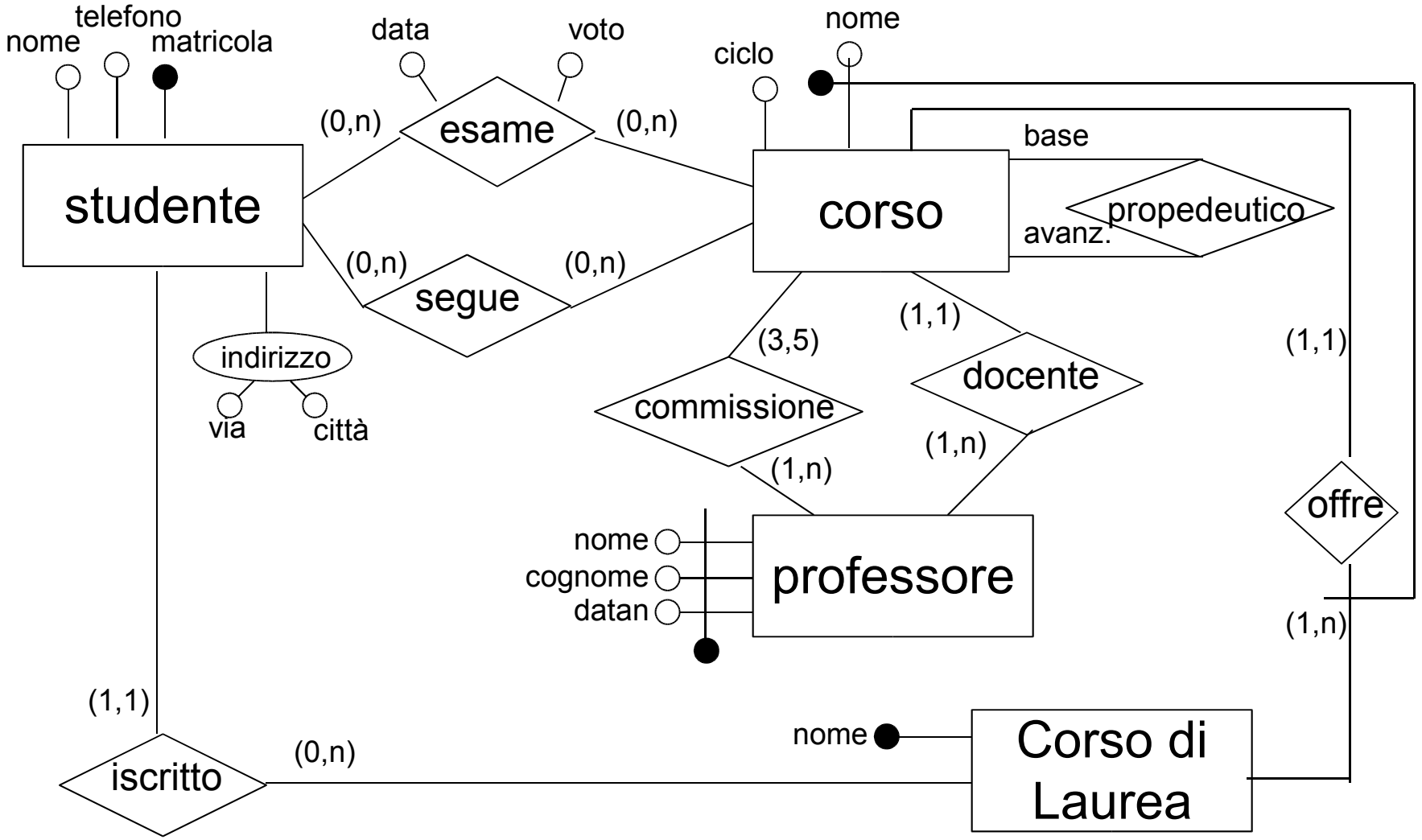
Identificatore esterno composto
una matricola è univoca solo all'interno della stessa Università

Identificatori: alcuni dettagli

- **Se E è identificata esternamente** attraverso l'associazione A, allora si ha **min-card(E,A) = max-card(E,A) =1**
- Se basta E1, tramite A, a identificare E, allora max-card(E1,A) =1; in caso contrario max-card(E1,A) = n
- Ogni entità deve avere almeno un identificatore, in generale può averne più di uno
- Alle volte si dice che E è un'**entità debole** se ha **solo identificatori esterni**, e **forte** se ha **solo identificatori interni**

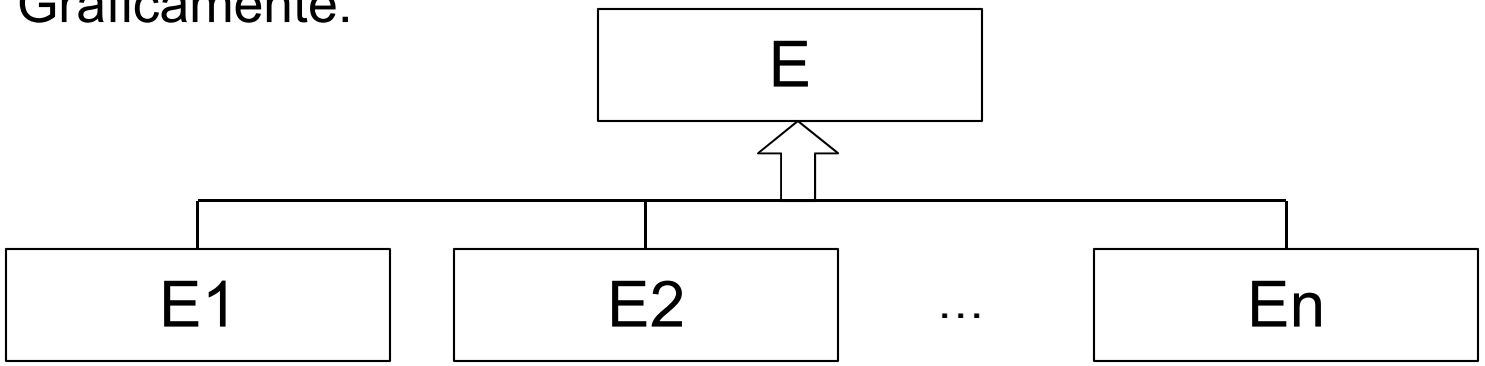
- È possibile evidenziare graficamente un'entità debole disegnando il rispettivo rettangolo con una doppia linea

Uno schema E-R (completo!)



Gerarchie di generalizzazione

- Un'entità E è una **generalizzazione** di un gruppo di entità E1, E2, ..., En se ogni istanza di E1, E2, ..., En è anche un'istanza di E
- Le entità E1, E2, ... En sono dette **specializzazioni** di E
- Graficamente:

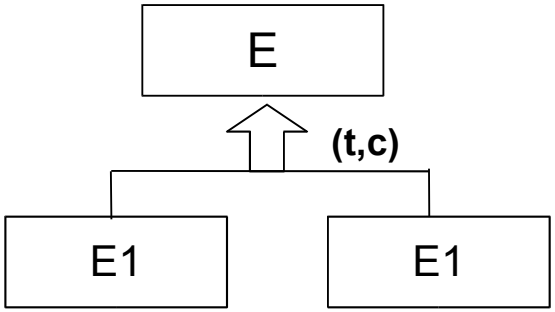


- Le proprietà di E sono ereditate da E1, E2, ..., En: **ogni Ei ha gli attributi di E e partecipa alle relazioni definite per E** (*non vano quindi replicati nello schema, sarebbe un errore!)*
- Per le gerarchie di generalizzazione va anche specificato il **tipo di copertura**

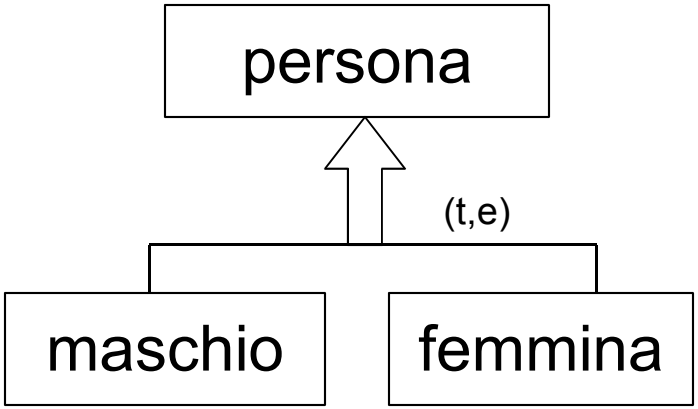
Proprietà di copertura

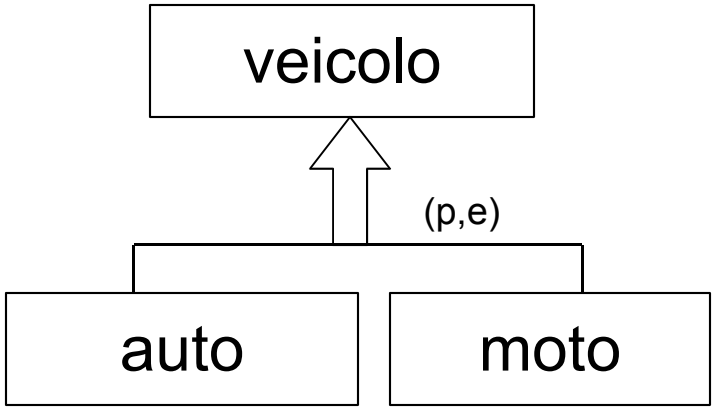
- La copertura può essere **totale** o **parziale**.
  - **Totale**: ogni istanza dell'entità generica deve essere necessariamente presente in almeno una delle sotto-entità
  - **Parziale**: può esistere una istanza dell'entità generica che non appartiene a nessuna delle sotto-entità

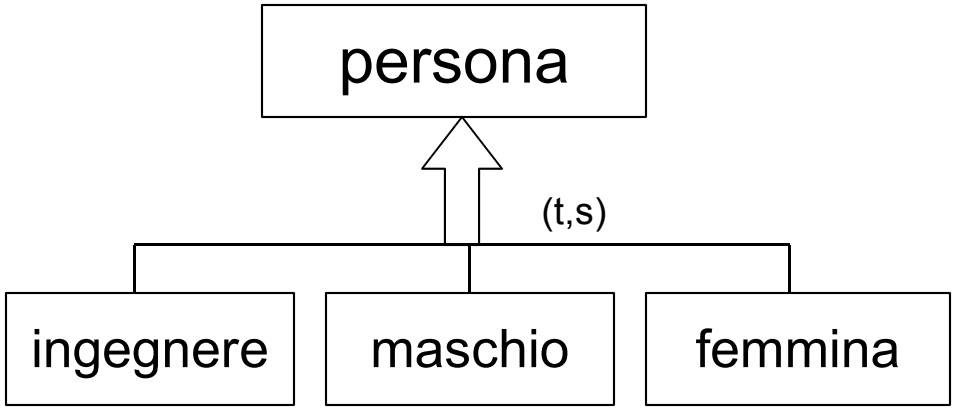
- La copertura può essere **esclusiva** o con sovrapposizione.
  - **Esclusiva**: ogni istanza dell'entità generica può essere presente al più in una delle sotto-entità
  - **Con sovrapposizione**: può esistere una istanza dell'entità generica che appartiene a più di una sotto-entità

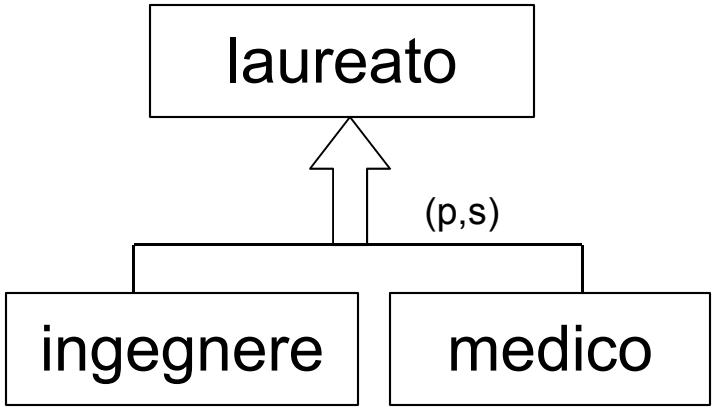


Proprietà di copertura - esempi





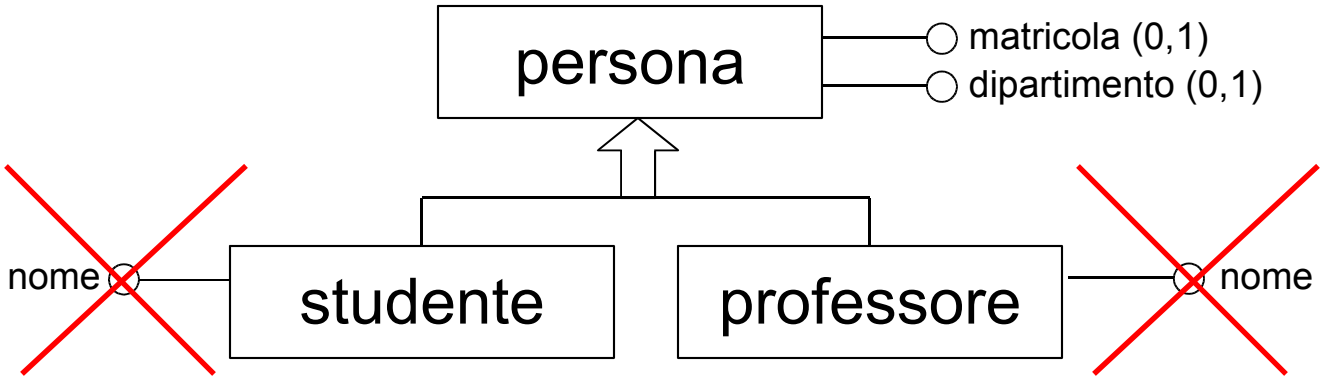




Ereditarietà delle proprietà

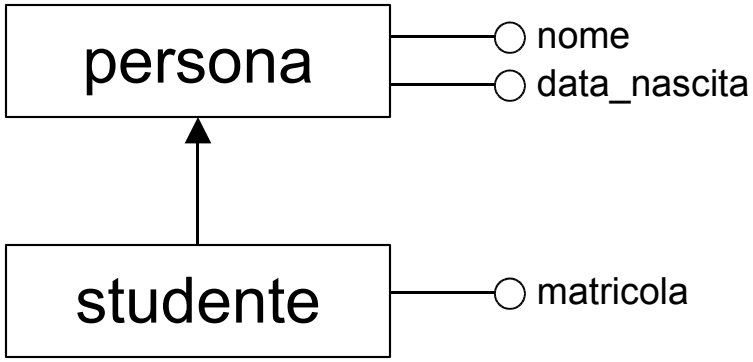
- Gli attributi vanno riferiti all'entità più generica in cui sono presenti obbligatoriamente
- Analogamente per le relazioni

Quindi così non va bene:



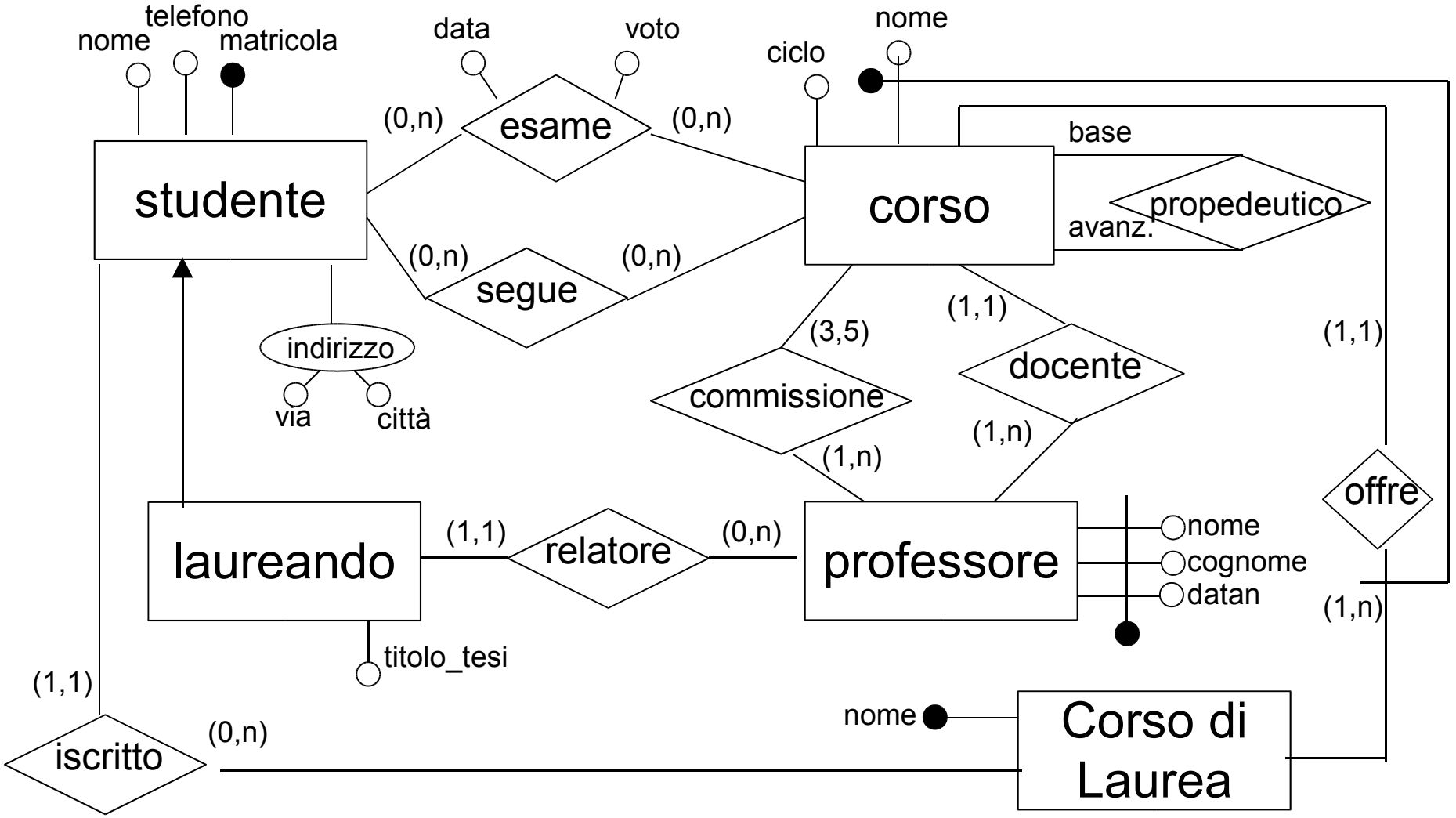
Subset

- È un caso particolare di gerarchia in cui si evidenzia una sola classe **specializzata**
  - Studente eredita le proprietà di Persona e in più ha la matricola
  - ogni Studente è anche una Persona



Non ha ovviamente senso parlare di tipo di copertura

Uno schema E-R (completo!)



Esercizi

Esercizio sulle gerarchie

- Le persone hanno CF, cognome ed età; gli uomini anche la posizione militare; 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) un numero di matricola; esistono persone che non sono né impiegati né studenti (ma i dettagli non ci interessano)

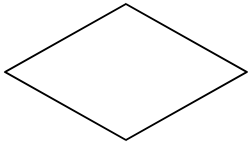
Esercizio sulla teoria

- I concetti sinora introdotti per il modello E/R possono essere modellati disegnando uno schema E/R (!)
  Ad esempio:
  - Ogni entità ha almeno un identificatore (interno o esterno)
  - Ogni associazione, in base al suo grado, è collegata a n entità
  - Ogni attributo ha un nome (univoco all'interno dell'entità o associazione a cui si riferisce)

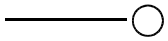
Notazione grafica (riassunto)

entità

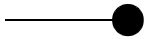
relazione



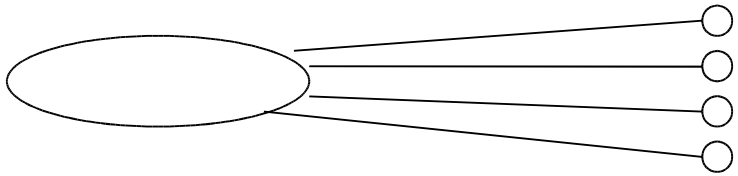
attributo



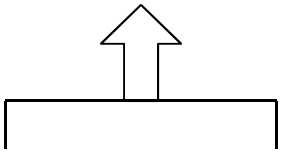
identificatore



attributo composto



gerarchia di generalizzazione



subset

Vincoli di cardinalità

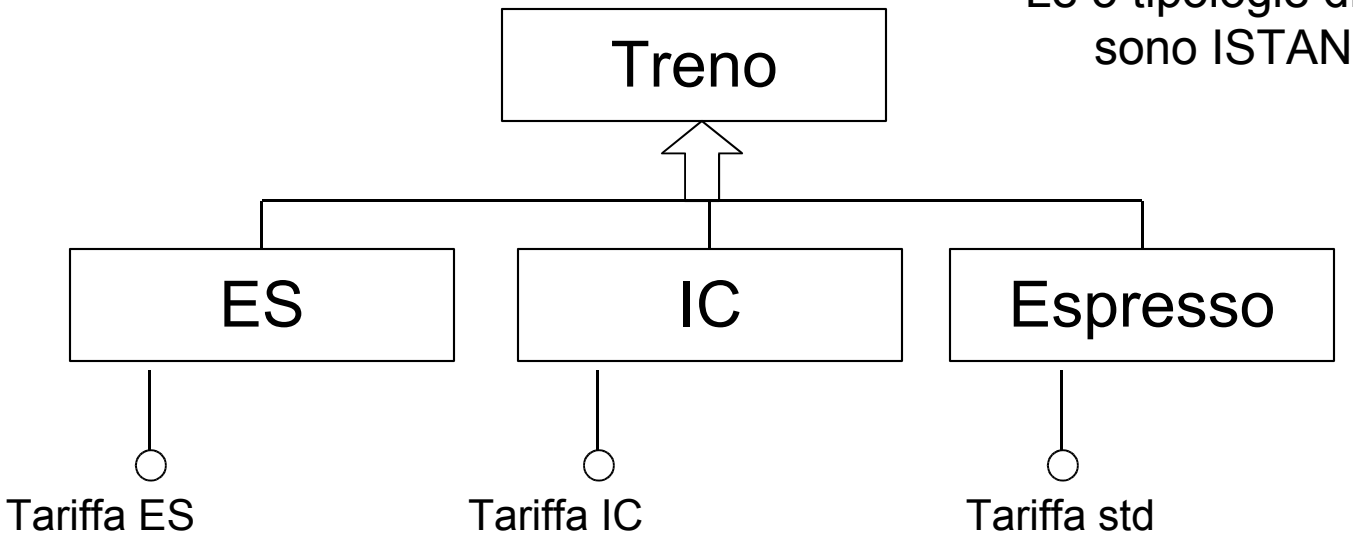
(min-card,max-card)

Esempio: Categorie dei treni

- *… le tariffe del treno sono: prezzo EuroStrar, supplemento InterCity e tutte le altre ….*

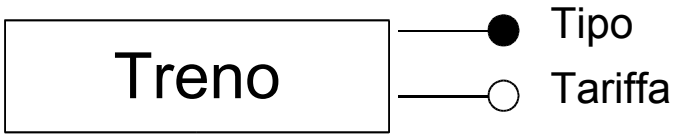
NOOO

Le 3 tipologie di treno sono ISTANZE



… istanze o gerarchia?

- La soluzione consiste nel **non introdurre la gerarchia** e nello specificare come identificatore il tipo di treno:



- In generale, attenzione a non prendere per tipologie (e quindi per specializzazioni di un’entità) quelle che sono solo istanze dell’entità

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

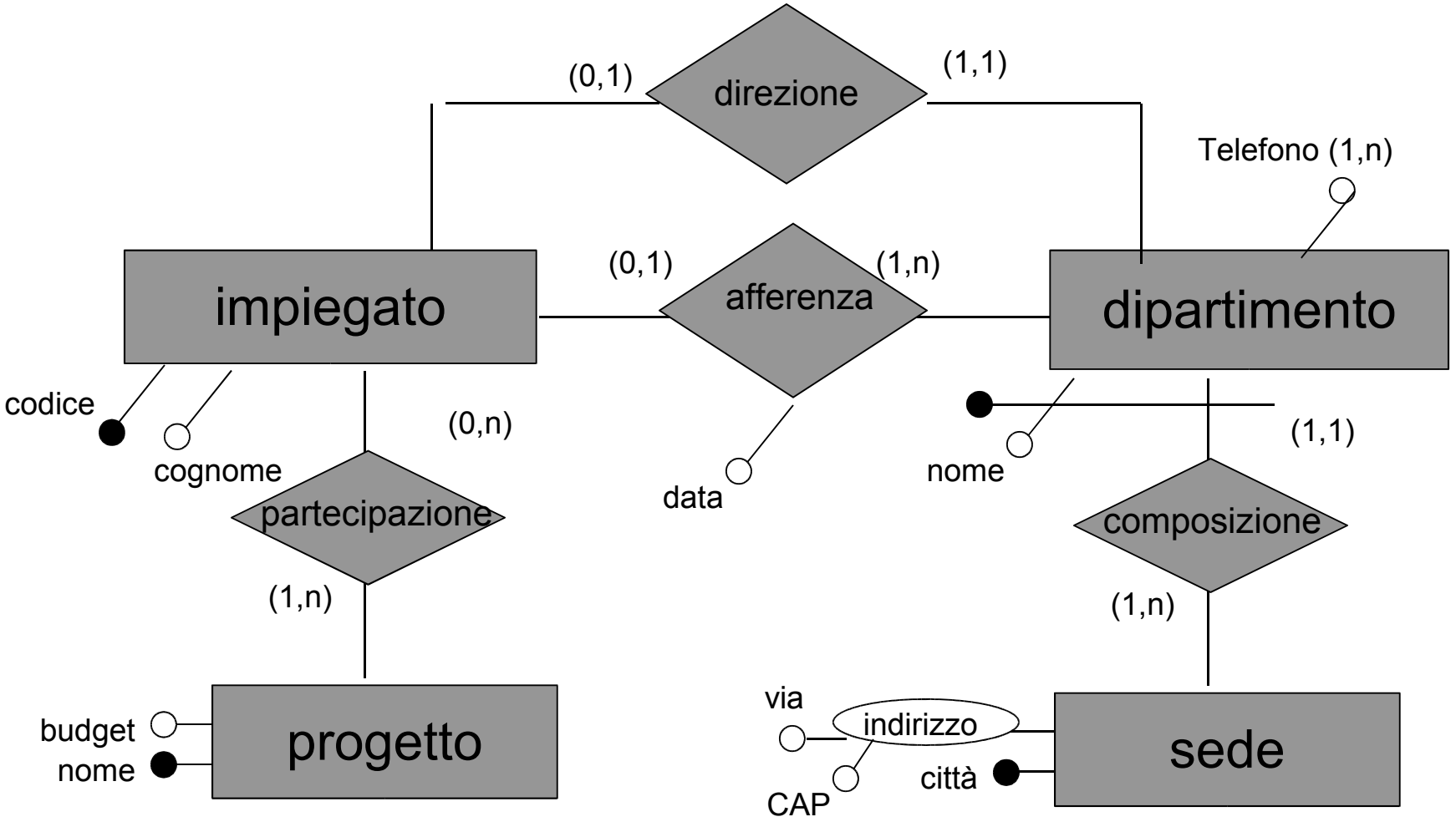


Tavola dei volumi

- 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

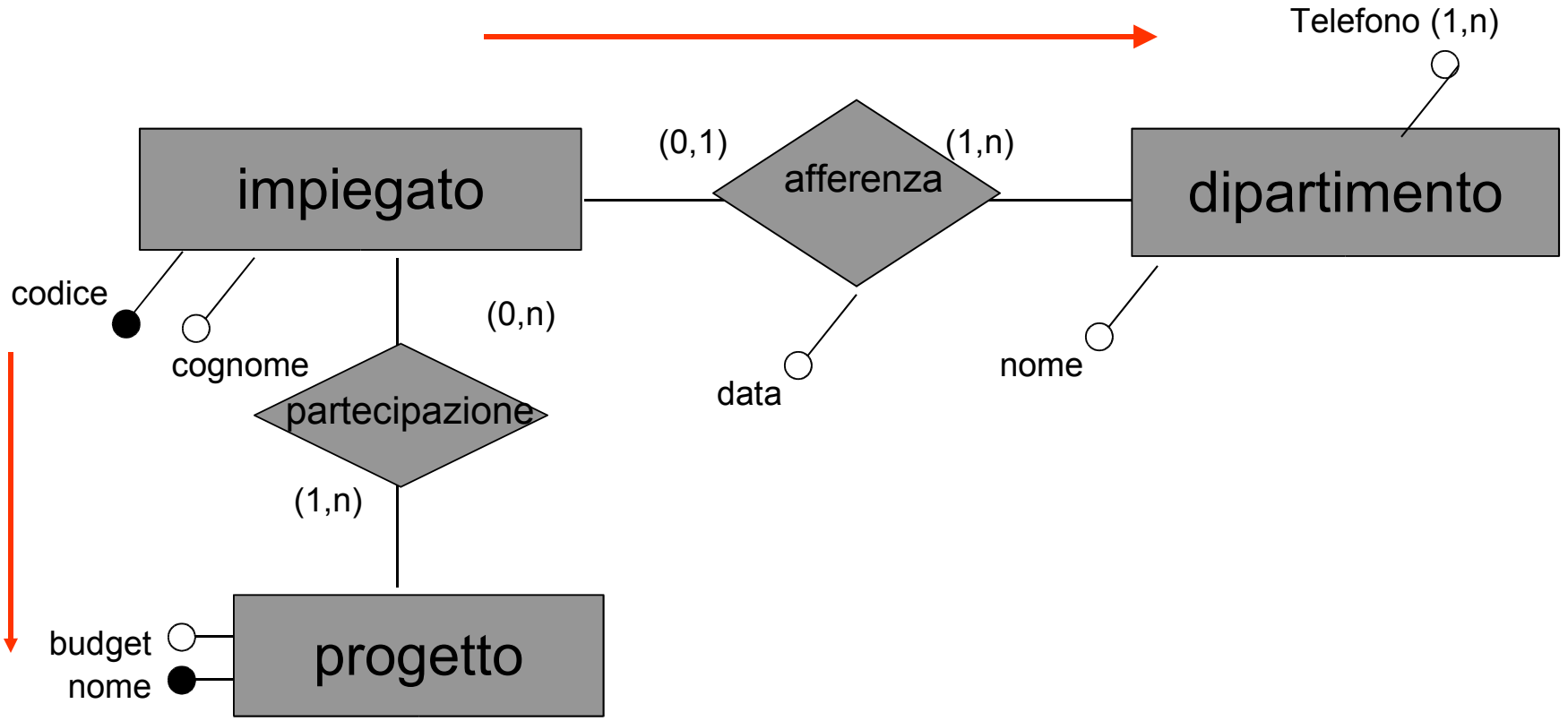


Tavola degli accessi

- 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

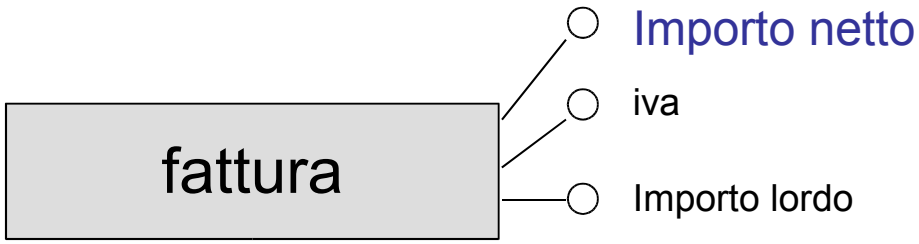
Analisi delle ridondanze

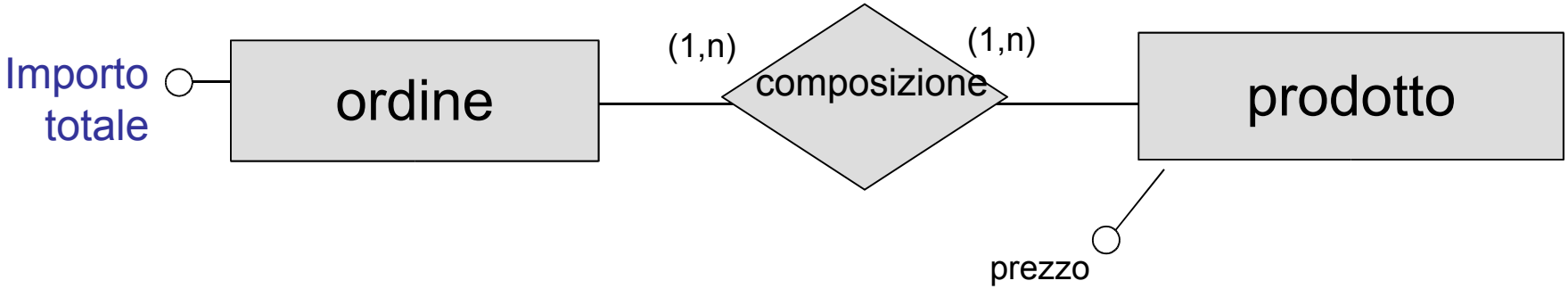
- Una **ridondanza** in uno schema E-R è ***una informazione significativa ma derivabile da altre***
- In questa fase si decide se eliminare le ridondanze eventualmente presenti o mantenerle (è quindi comunque importante averle individuate in fase di progettazione concettuale!)

- Se si mantiene una ridondanza
  - si semplificano alcune interrogazioni, ma
  - si appesantiscono gli aggiornamenti
  - si occupa maggior spazio

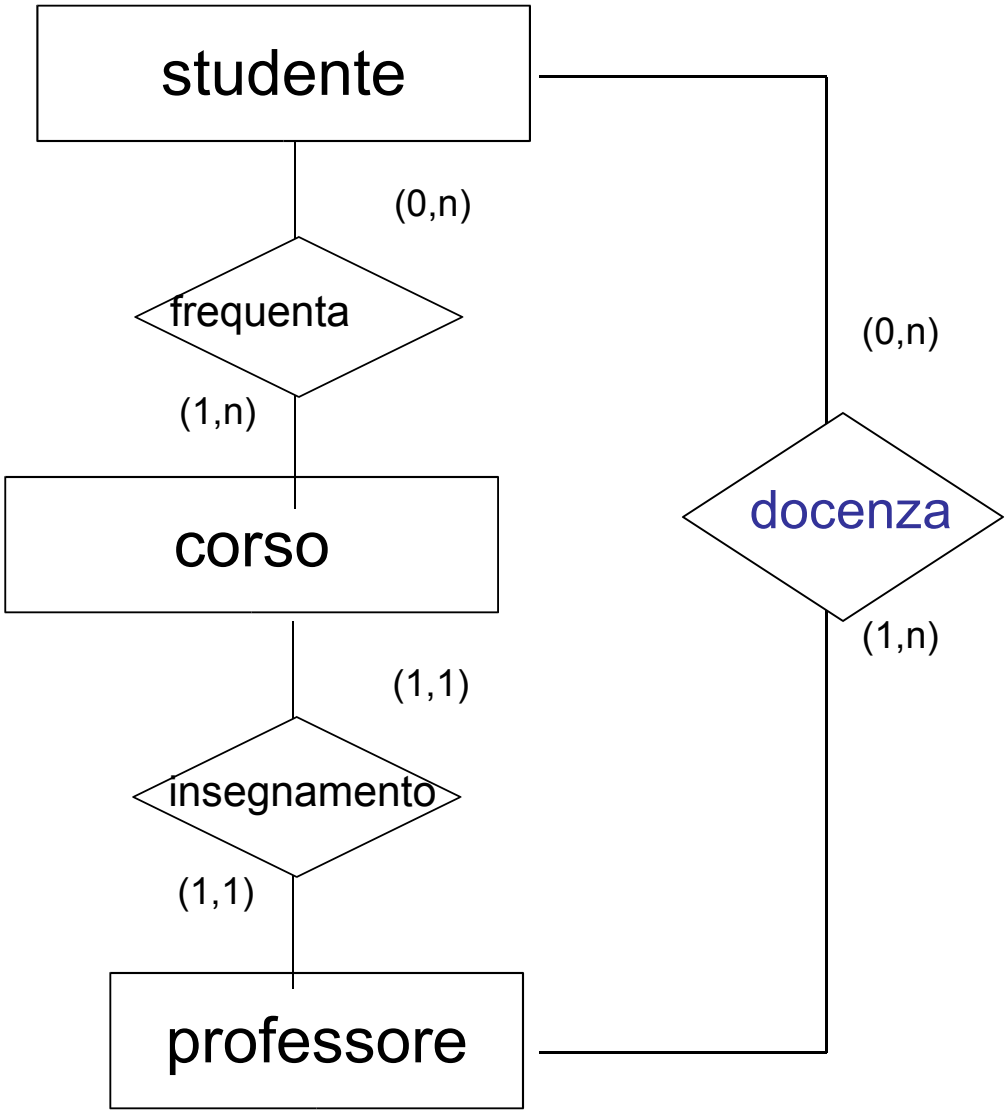
- Le possibili ridondanza riguardano
  - Attributi derivabili da altri attributi
  - Relazioni derivabili dalla composizione di altre relazioni (presenza di cicli)

Attributi derivabili





Associazioni ridondanti

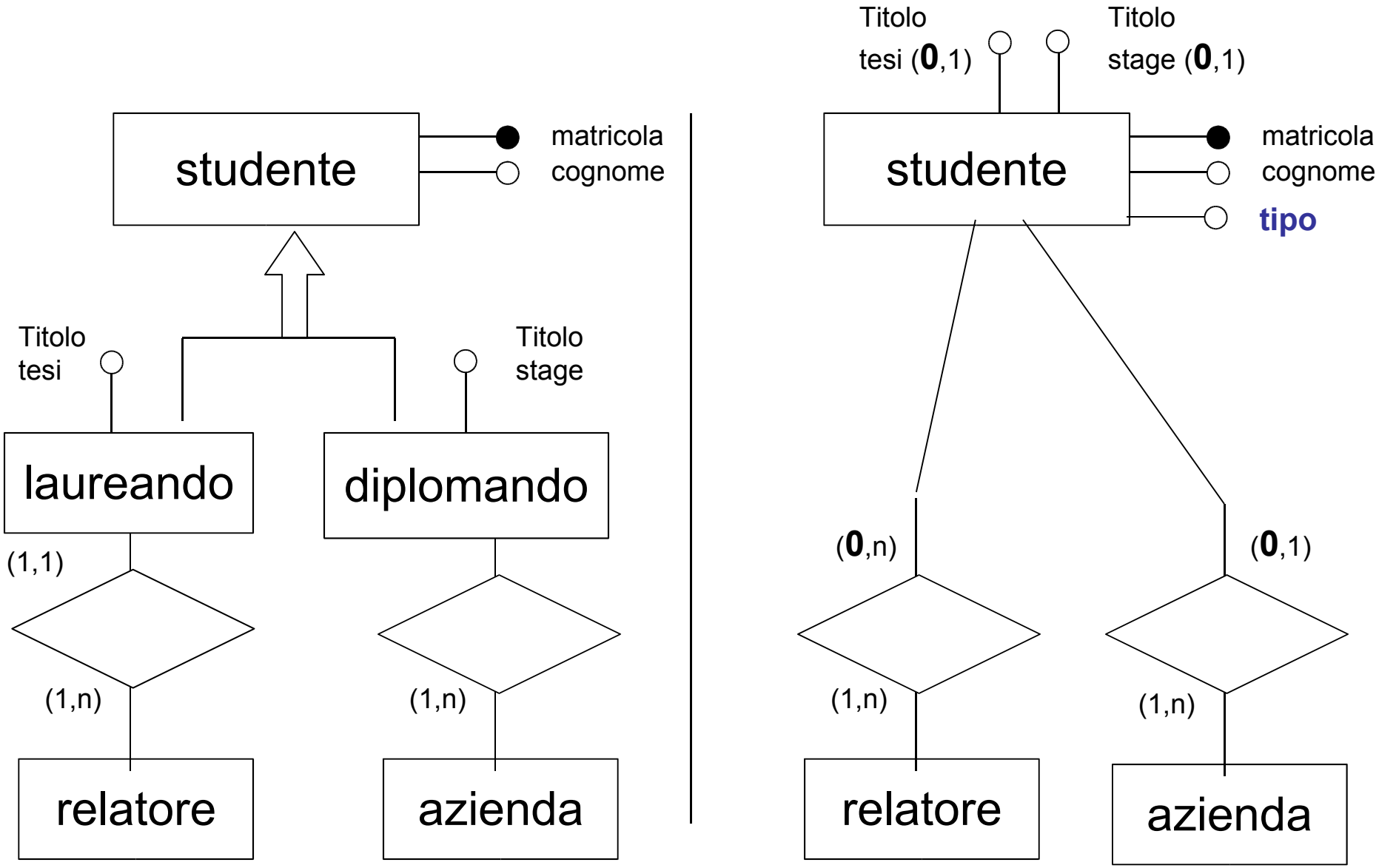


Gerarchie

- 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



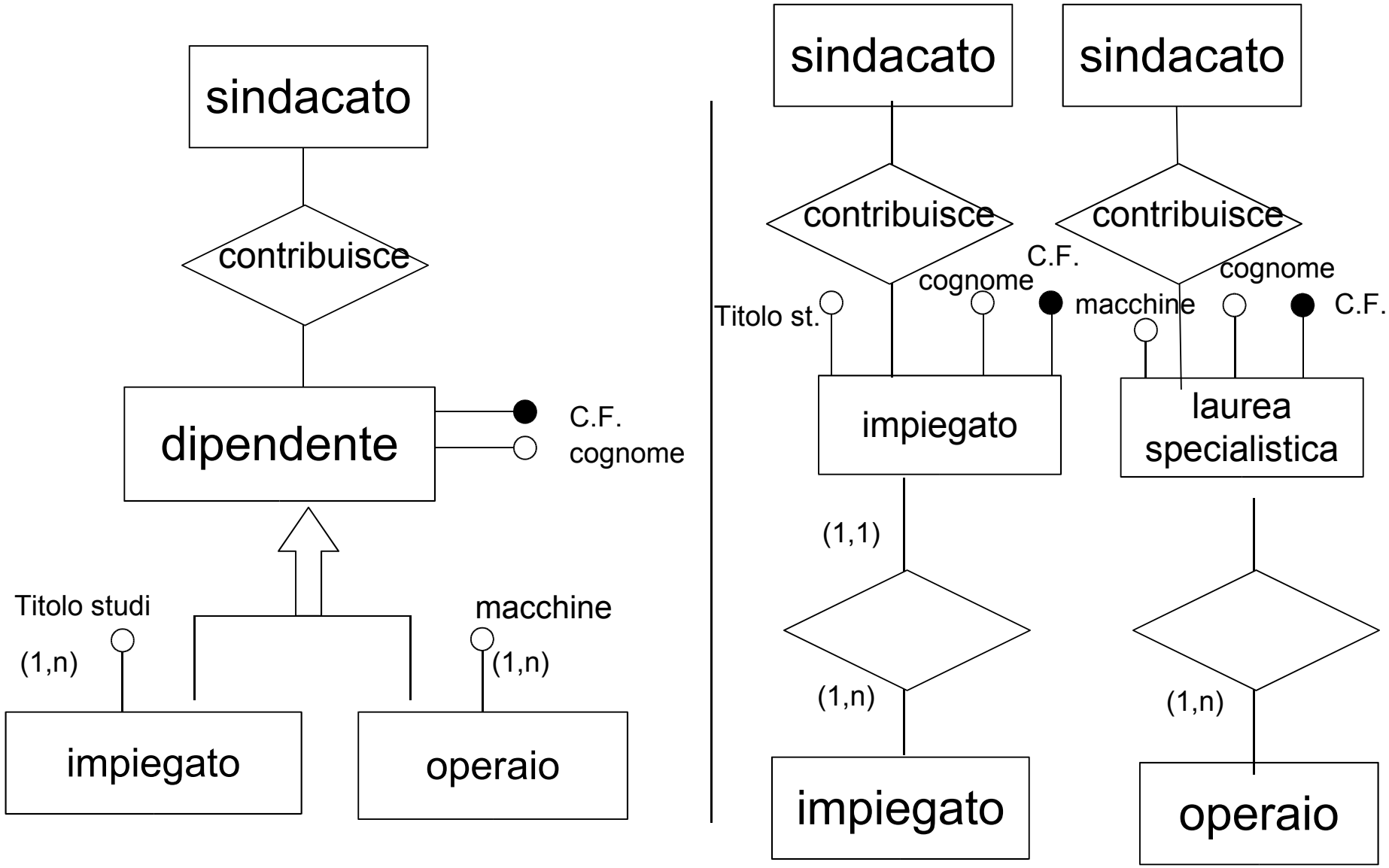
Collasso verso l'alto

- **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

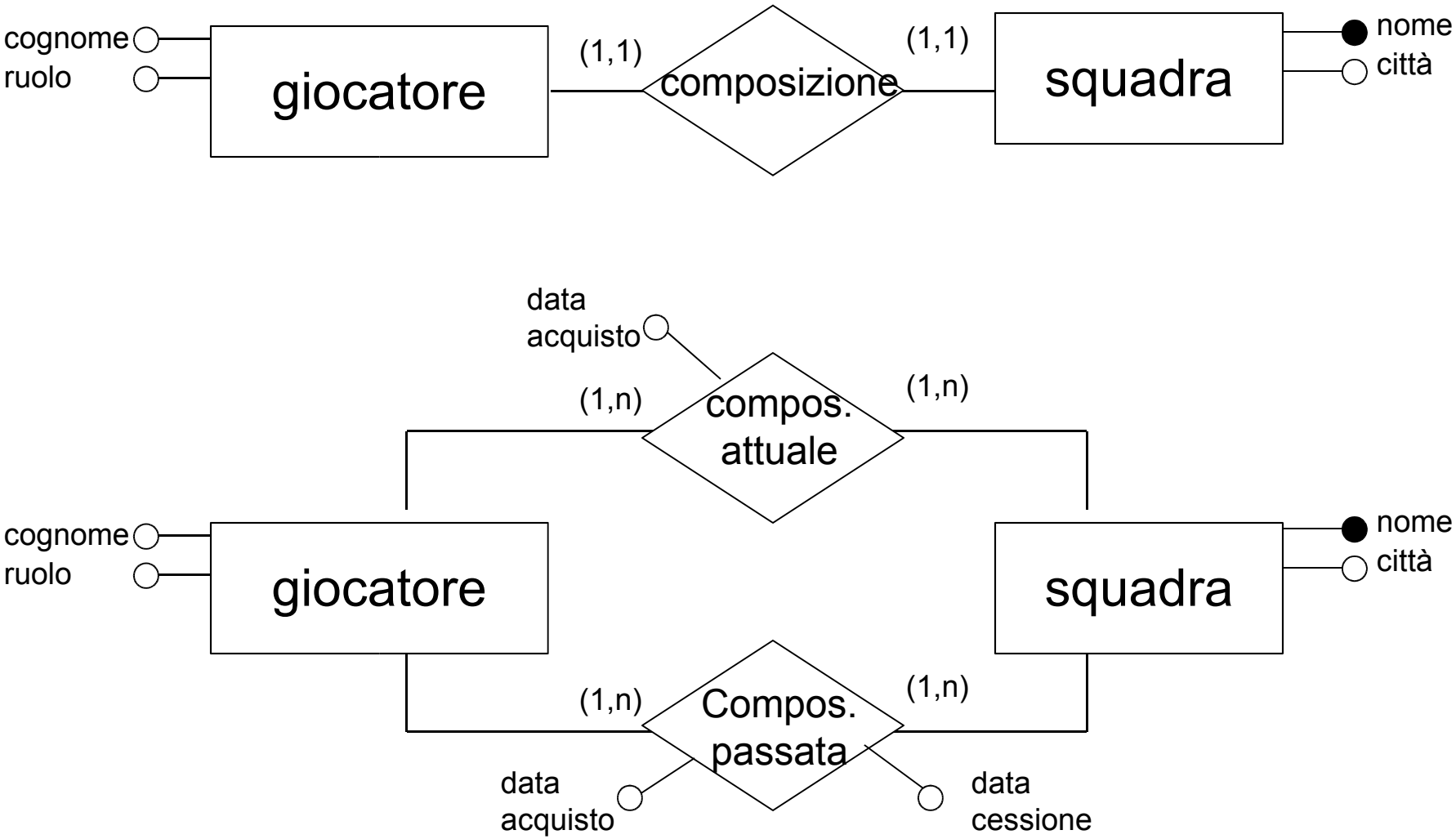
Partizionamenti e accorpamenti

- è possibile ristrutturare lo schema accorpando 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

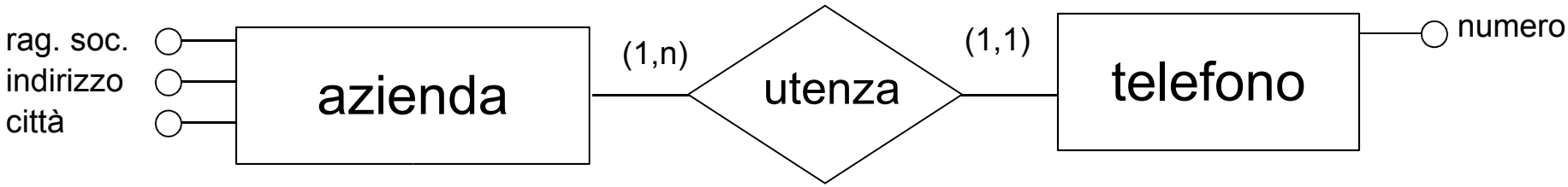
Si separano gli attributi in gruppi omogenei

Partizionamento orizzontale di relazioni

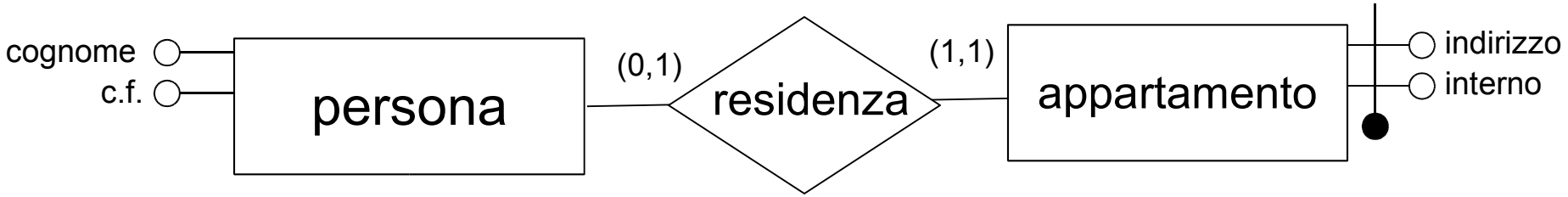


Eliminazione di attributi multivalore

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**

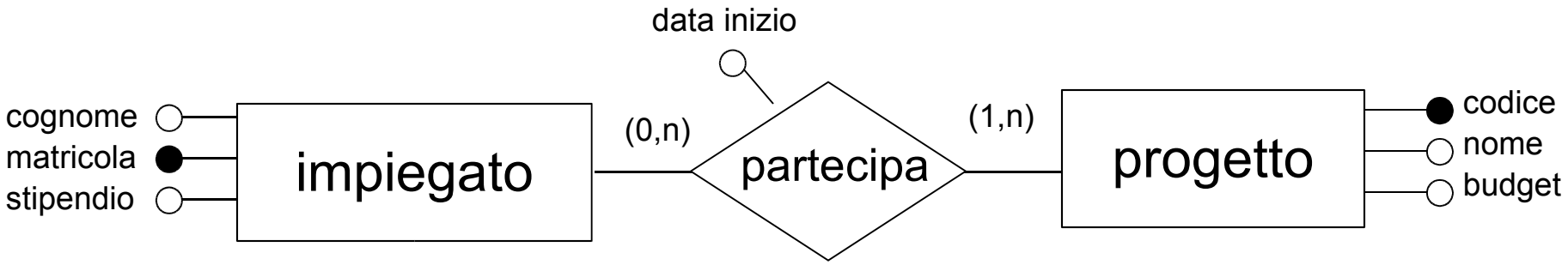
Traduzione delle entità

Persona(<u>CF</u>, 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

Foreing key

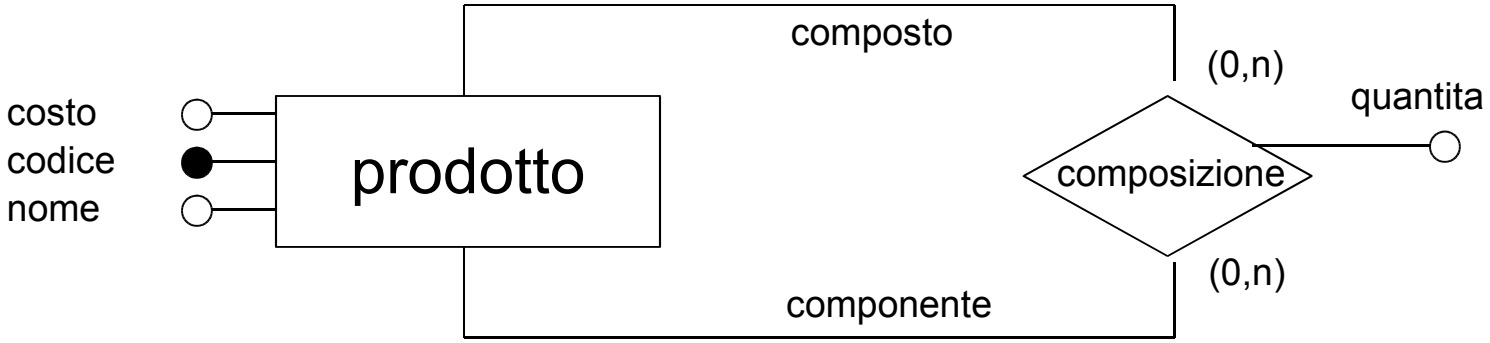
- Non è ovviamente necessario mantenere per gli attributi chiave della tabella che traduce la relazione gli stessi nomi delle chiavi primarie referenziate, ma conviene usare nomi più espressivi

- 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

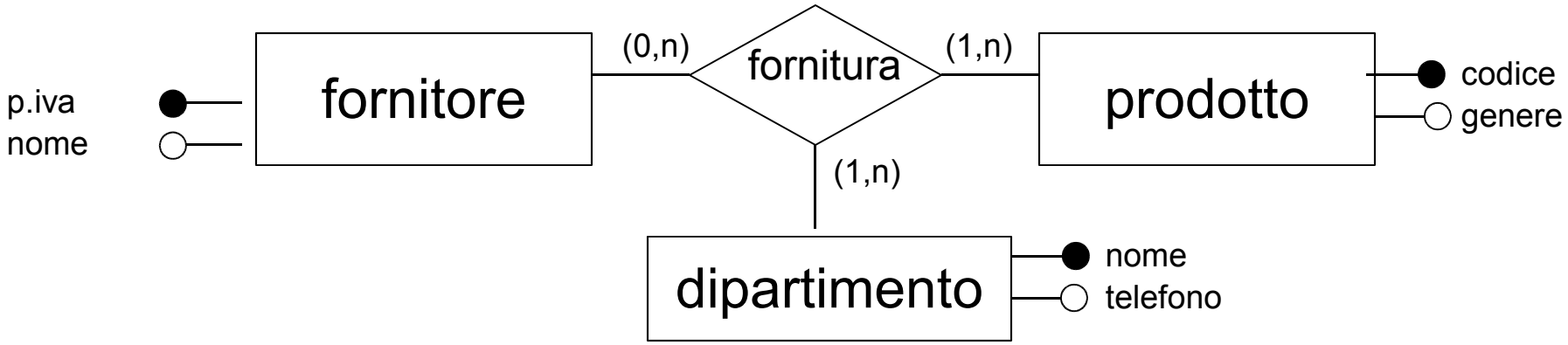
- In questo caso i nomi degli attributi che formano la chiave primaria della relazione si possono derivare dai **ruoli** presenti nei rami dell'associazione



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

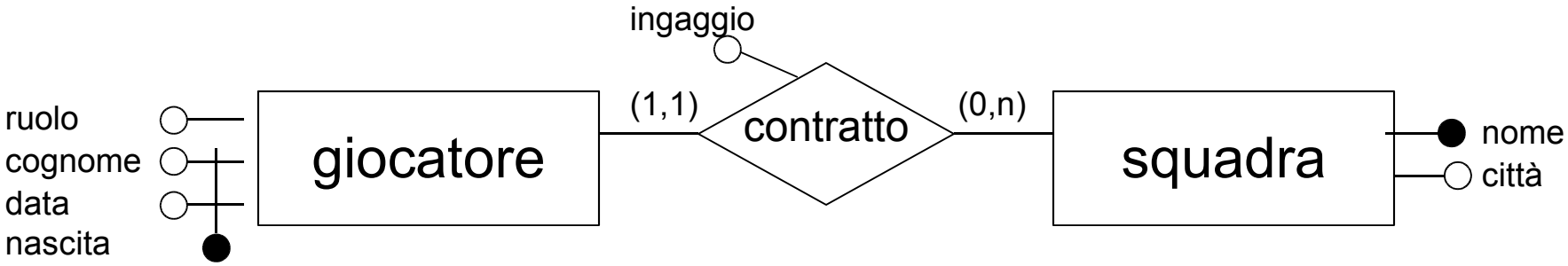
Associazioni n-arie molti a molti

- 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

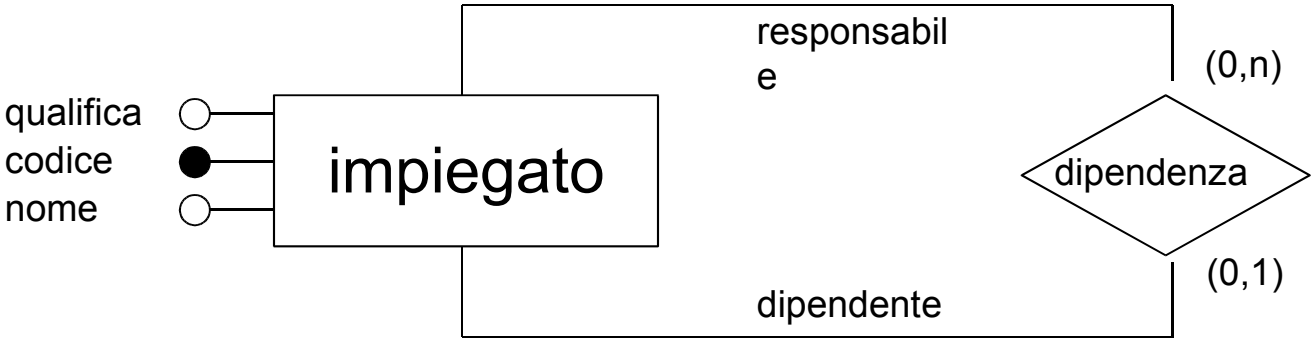
- Poiché un giocatore ha un contratto con una sola squadra, nella relazione Contratto un giocatore non può apparire in più tuple
- Si può pertanto pensare anche ad una soluzione più compatta, facente uso di 2 sole relazioni

  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



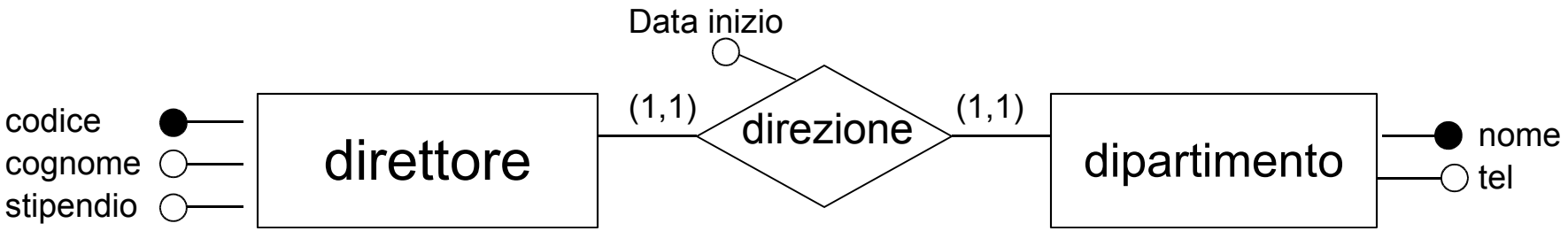
Impiegato(<u>Codice</u>, Nome, Qualifica, Responsabile*)
FK: Responsabile REFERENCES Impiegato

1 tabella

Impiegato(<u>Codice</u>, Nome, Qualifica)
Dipendenza(<u>Dipendente</u>, Responsabile)
FK: Dipendente REFERENCES Impiegato
FK: Responsabile REFERENCES Impiegato

2 tabelle

Relazioni uno a uno



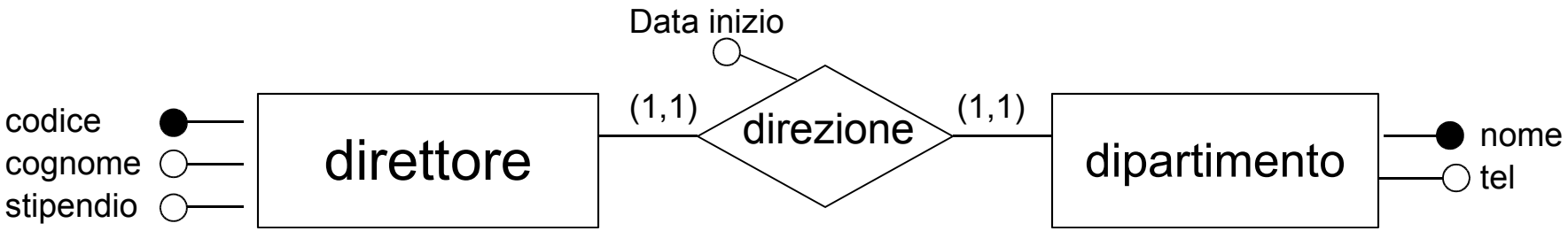
3 tabelle

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



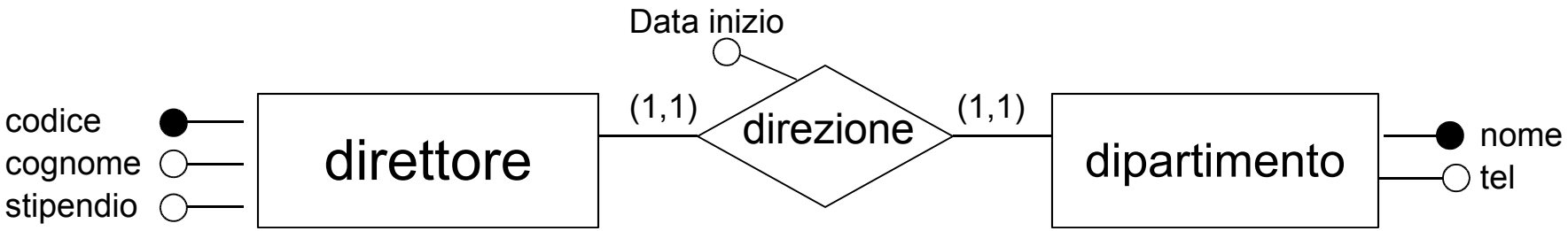
2 tabelle

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)

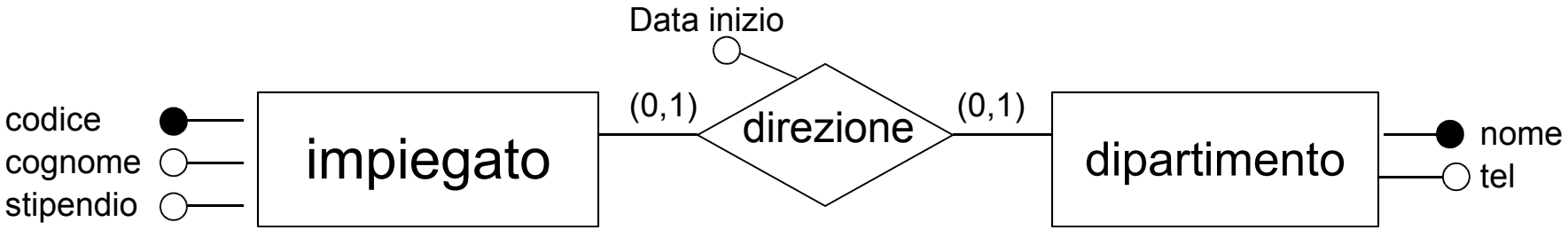
oppure

Direttore(Codice, Cognome, Stipendio)
Dipartimento(Nome, Sede, Telefono, Direttore, DataInizio)
FK: Direttore REFERENCES Direttore

Relazioni ad anello uno a molti

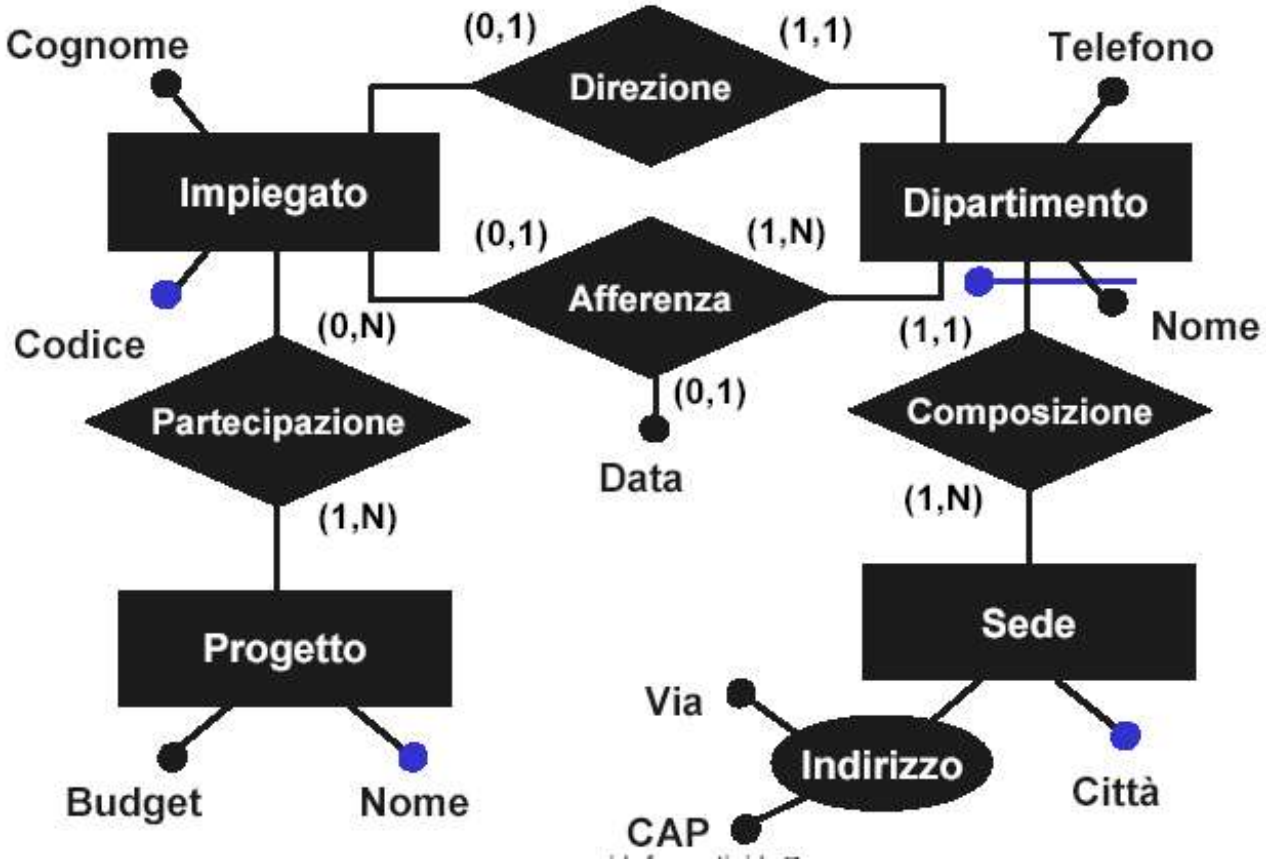
- In linea di principio la traduzione con una sola tabella non andrebbe qui considerata, in quanto corrisponde a un accorpamento di entità, oggetto della fase di ristrutturazione.

- Se min-card(E1,R) = min-card(E2,R) = 1 si avranno due chiavi, entrambe senza valori nulli (la chiave primaria è “la più importante”)
- Se min-card(E1,R) = 0 e min-card(E2,R) =1 la chiave derivante da E1 ammetterà valori nulli, e la chiave primaria si ottiene da E2
- Se min-card(E1,R) = min-card(E2,R) = 0 entrambe le chiavi hanno valori nulli, quindi si rende necessario introdurre un codice



impDip(<u>CodiceImpDip</u>, CodiceImp*, …, Dipartimento*, …, DataInizio*)

Esempio di riferimento



Esempio: schema logico relazionale

- Per le entità E che partecipano a relazioni sempre con max-card(E,R) = n la traduzione è immediata:

  Sede(<u>Città</u>, Via, CAP)

  Progetto(<u>Nome</u>, Budget)

- Anche la relazione Partecipazione si traduce immediatamente:

  Partecipazione(<u>Impiegato</u>, <u>Progetto</u>)

- L'entità Dipartimento si traduce importando l'identificatore di Sede e inglobando l'associazione Direzione

  Dipartimento(<u>Nome</u>, 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(<u>Codice</u>, Cognome, Dipartimento*, Data*)

SQL

- SQL: Structured Query Language - linguaggio di definizione e manipolazione di dati

  Select AttrExpr [[as] Alias]{, AttrExpr [[as] Alias]}
  From TableName [[as] Alias]{, TableName [[as] Alias]}
  [**Where** condition]

- Join (interno):

  produce un risultato nel quale le righe sono tutte e sole quelle ottenibili dalle righe delle due tabelle di origine, in cui i valori dei campi in comune sono uguali (*campi di join, cioè di congiunzione*)

Query di selezione (1)

Sia data la seguente base di dati:

Cliente(CF, Nome, Cognome, Indirizzo, ….)
Prodotto(Codice, Marca, Modello)
Vendita(CodiceProd, CF, Prezzo, Data)

Scrivere il codice SQL per le seguenti query di selezione

Estrarre dalla base di dati l'elenco di tutti i clienti:

SELECT * FROM cliente; ← *con * si estraggono tutti i campi della tabella*

Query di selezione (2)

Estrarre dalla base di dati il nome e cognome del cliente con CF uguale a "AAABBB123456789G":

SELECT nome,cognome FROM cliente **WHERE CF= "AAABBB123456789G**";

Estrarre dalla base di dati il nome, cognome e indirizzo dei clienti della provincia di Forlì

SELECT nome,cognome,indirizzo FROM cliente WHERE provincia= "FO";

Query che coinvolgono la selezione da più tabelle: JOIN

Estrarre dalla base di dati l'elenco dei clienti che hanno effettuato almeno un acquisto:

SELECT cliente.CF,cliente.cognome FROM cliente**, vendita**

WHERE **cliente.CF=vendita.CF**; ← *riga di join*

Estrarre dalla base di dati l'elenco dei prodotti acquistati dal Sig. Rossi

SELECT prodotto.codice,prodotto.modello,vendita.prezzo FROM cliente, vendita,prodotto

WHERE **cliente.CF=vendita.CF AND prodotto.codice=vendita.codprodotto** ← *riga di join*

AND cliente.cognome="Rossi";

Estrarre dalla base di dati l'elenco dei clienti che hanno acquistato il prodotti di codice 520

SELECT cliente.cognome,cliente.nome FROM cliente, vendita, prodotto

WHERE **cliente.CF=vendita.CF AND prodotto.codice=vendita.codprodotto** ← *riga di join*

AND prodotto.codice=520;