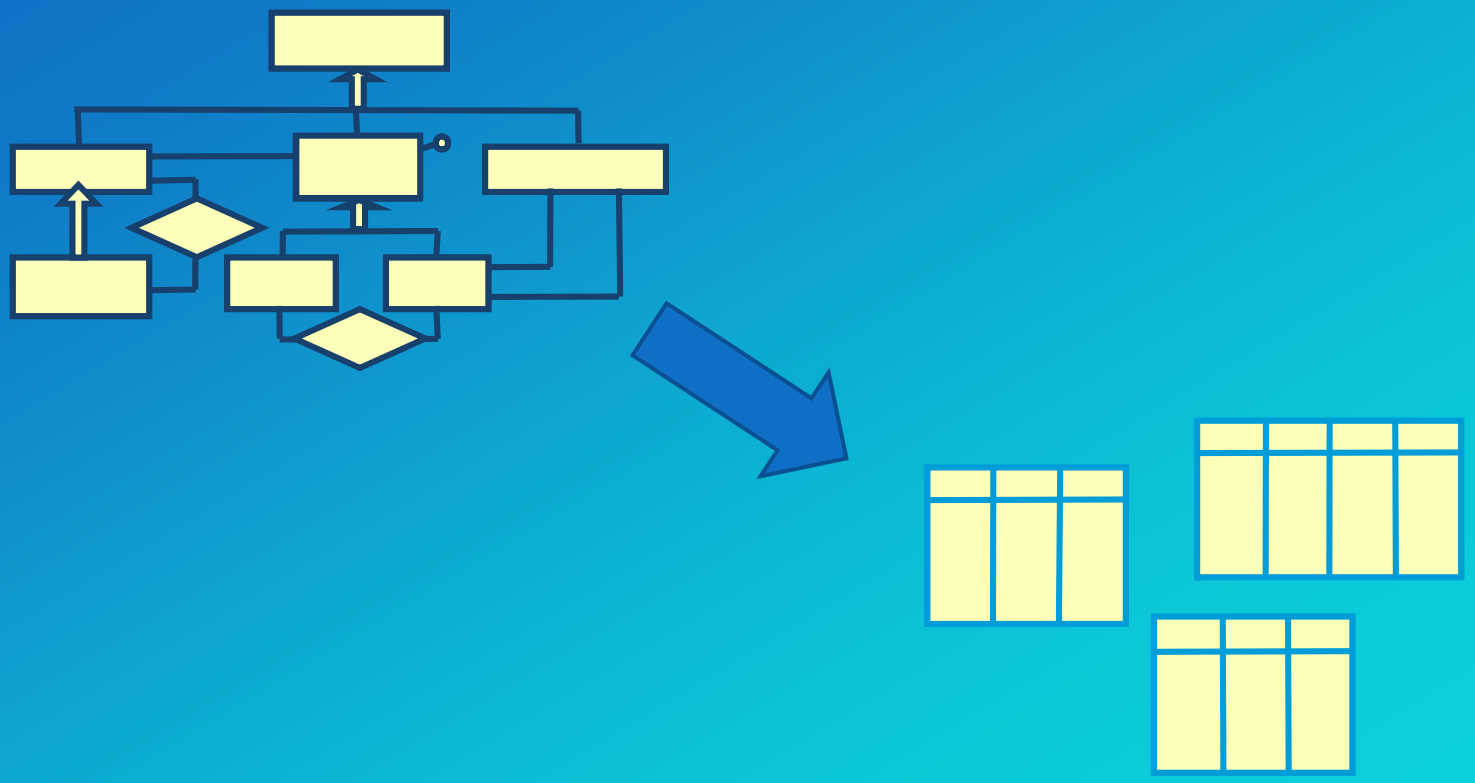
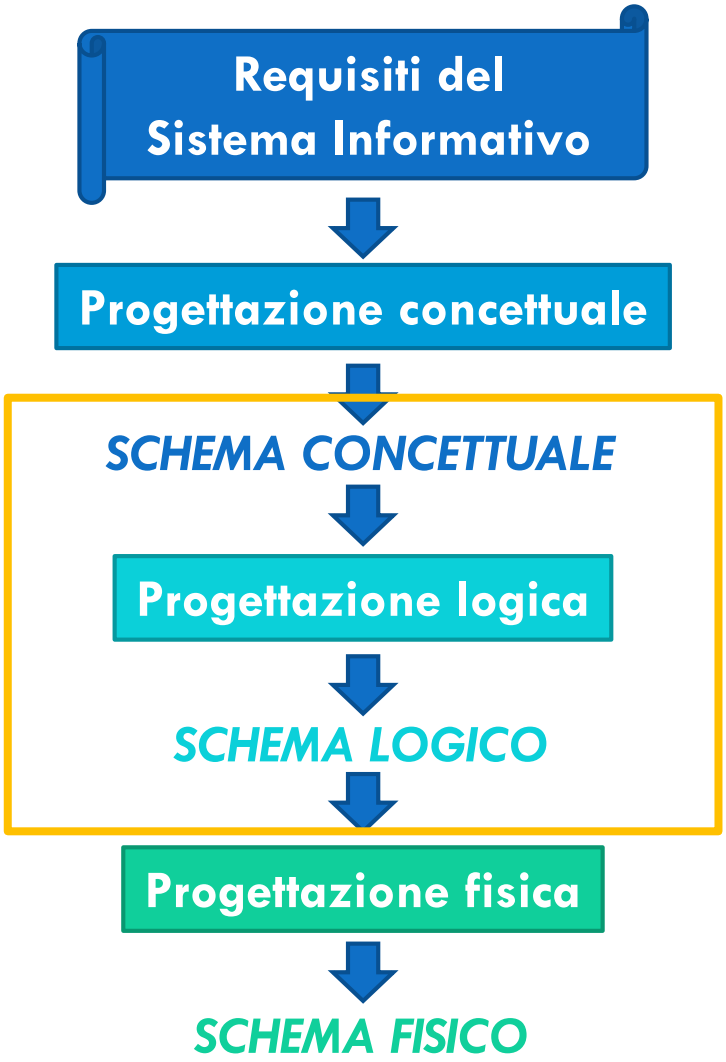


Progettazione logica

Annalisa Franco, Dario Maio
Università di Bologna

Il secondo passo…



Progettazione logica

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

- L'efficienza è legata alle prestazioni, ma poiché queste non sono valutabili precisamente, né a livello concettuale né a livello logico, si ricorre all'impiego di indicatori semplificati.

Progettazione logica "fedele" = equivalenza

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

DB_{rel} e DB_{conc} sono equivalenti se $SL(DB_{conc}) = SL(DB_{rel})$

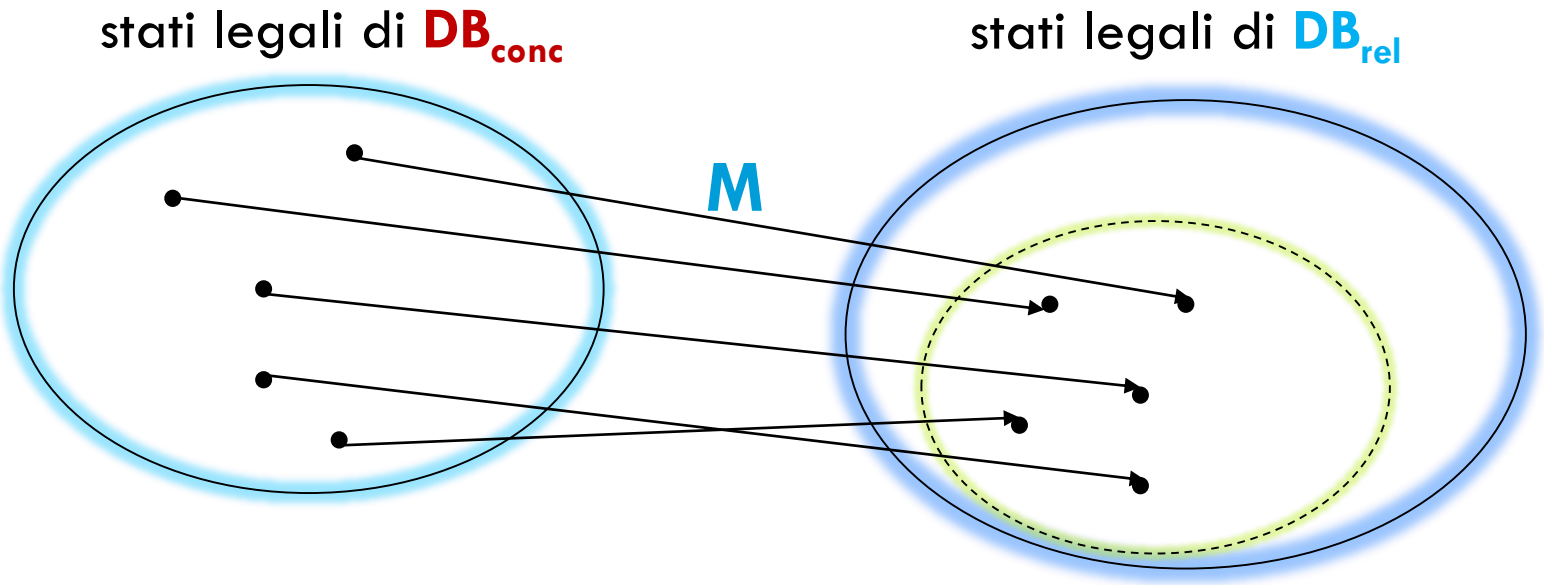
Progettazione che preserva l'informazione (1)

- Si consideri una progettazione che traduce un dato schema concettuale DB_{conc} in uno schema logico-relazionale DB_{rel}.
- Questa attività di progettazione può essere vista, a livello astratto, come la definizione di un **mapping M** che spiega come trasformare ogni stato legale db_{conc} di DB_{conc} in un corrispondente stato db_{rel} di DB_{rel}.

- La progettazione **preserva l'informazione se M è totale e iniettiva**:
  - (***totale***) per ogni stato db_{conc} di DB_{conc} esiste uno stato db_{rel} di DB_{rel} tale che $M(db_{conc}) = db_{rel}$;
  - (***iniettiva***) non esistono due stati $db1_{conc}$ e $db2_{conc}$ tali che $M(db1_{conc}) = M(db2_{conc})$.

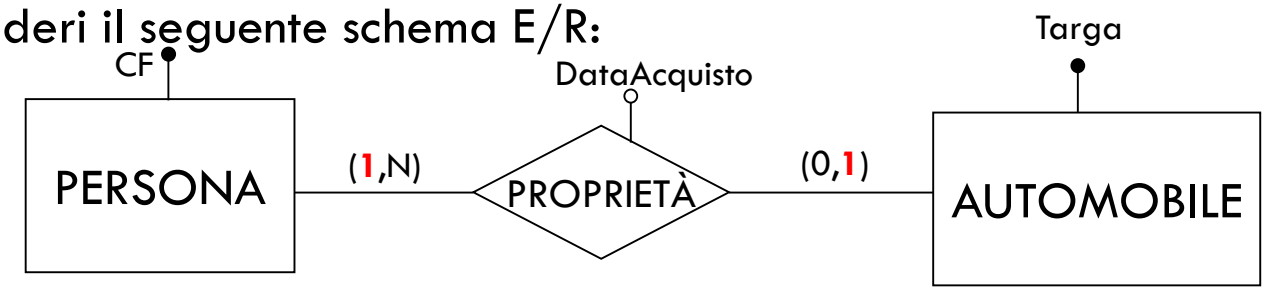
Progettazione che preserva l'informazione (2)

- Preservare l'informazione:
  - la definizione intuitivamente asserisce che lo schema relazionale può contenere i dati dello schema E/R (***totalità***) e che si può "ritornare indietro" (***iniettività***).



Perché ciò non basta

- Si consideri il seguente schema E/R:



e lo schema relazionale:

PERSONE(<u>CF</u>)
AUTOMOBILI(<u>Targa</u>)
PROPRIETÀ(<u>CF</u>, <u>Targa</u>, DataAcquisto)
FK: CF REFERENCES Persone
FK: Targa REFERENCES Automobili

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

PERSONE

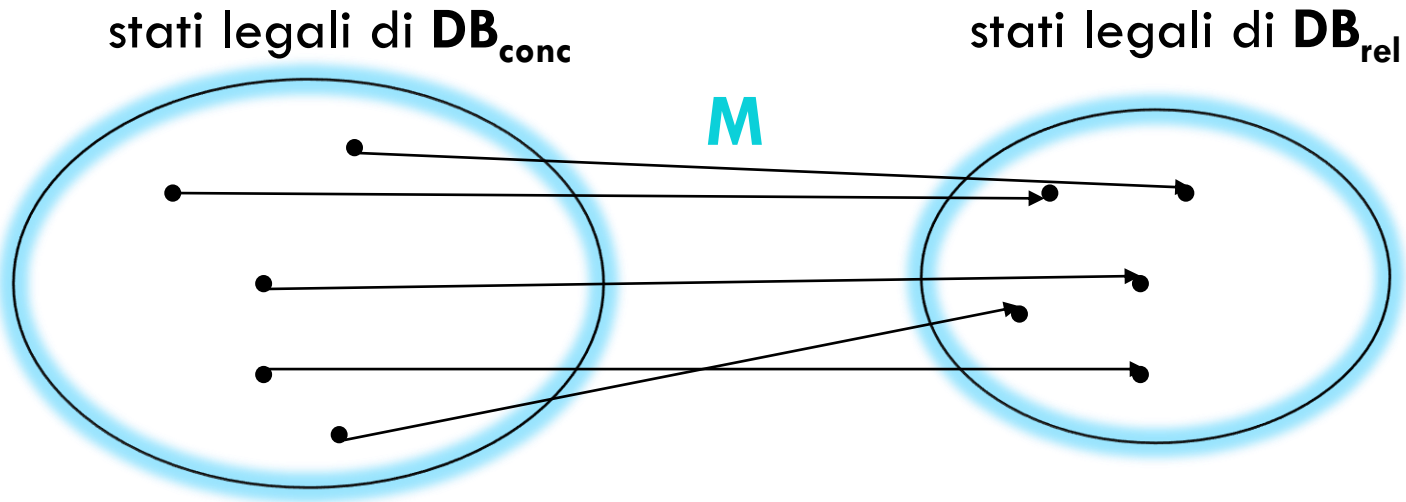
CF
BNCGRG78L21A944Z
RSSNNA78A53A944N
VRDMRC79H20F839U

PROPRIETÀ

CF	Targa	DataAcquisto
BNCGRG78L21A944Z	CT 001 MJ	12/08/2004
RSSNNA78A53A944N	CT 001 MJ	15/07/2003

Progettazione che garantisce l'equivalenza

- Diciamo che la progettazione garantisce l'equivalenza se:
  - preserva l'informazione e
  - per ogni stato legale db_{rel} di DB_{rel} esiste uno stato legale dbconc di DB_{conc} tale che $M(db_{conc}) = db_{rel}$.
- La definizione intuitivamente asserisce che esiste una biiezione tra gli insiemi di stati legali.



Come agire in pratica?

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

Fasi della progettazione logica

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

Fase di ristrutturazione

- Si pone l'obiettivo di semplificare la traduzione e "ottimizzare" le prestazioni.
- Per confrontare tra loro diverse alternative bisogna conoscere, almeno in maniera approssimativa, il "carico di lavoro", ovvero:
  - le principali **operazioni** che la base dati dovrà supportare;
  - i **"volumi" dei dati** in gioco.

Regola 80-20: il 20% delle operazioni produce l'80% del carico.

- Gli indicatori che deriviamo considerano due aspetti
  - spazio: numero di istanze (di entità e associazioni) previste;
  - tempo: numero di istanze visitate durante un'operazione.

Schema di riferimento

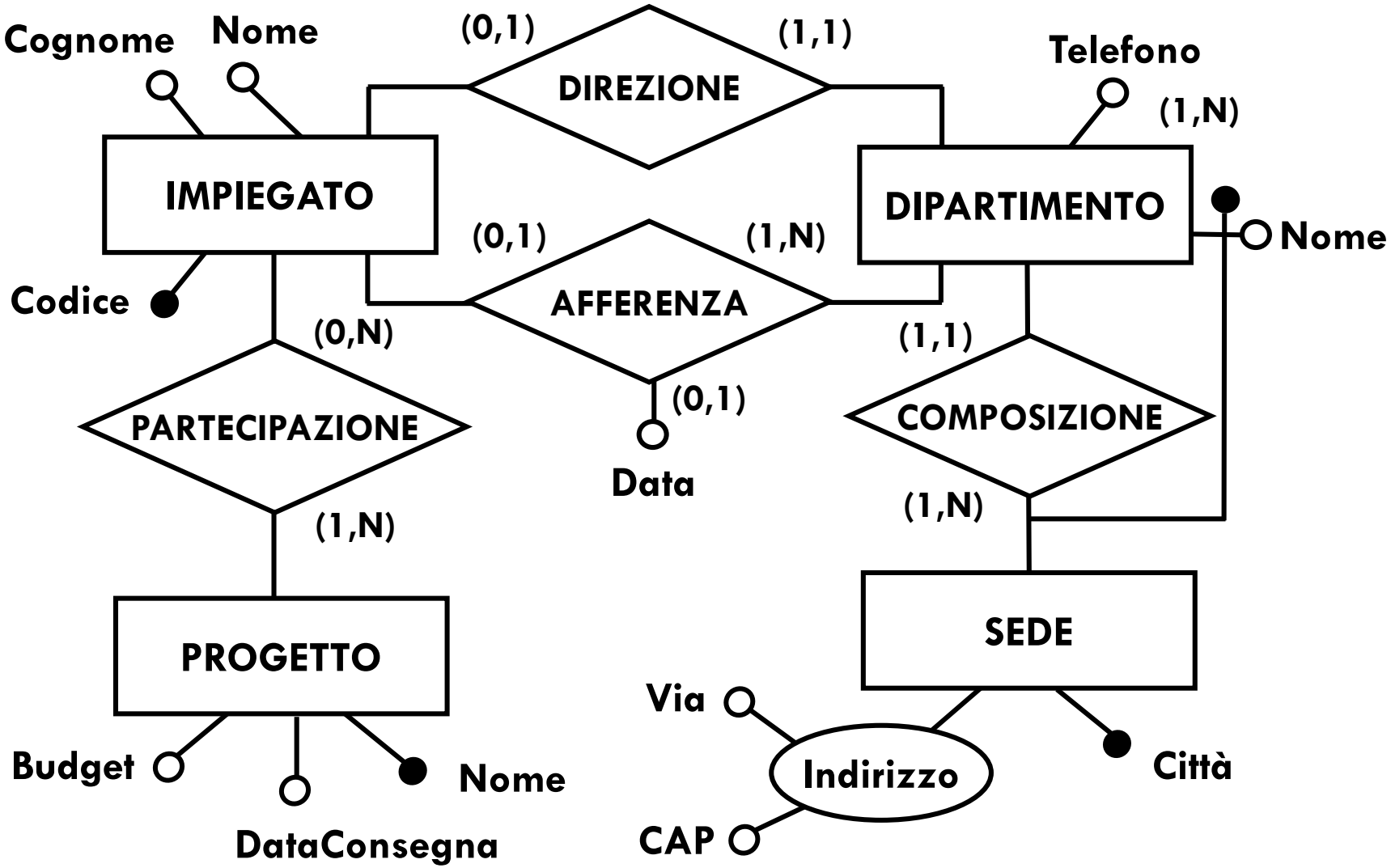


Tavola dei volumi

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

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

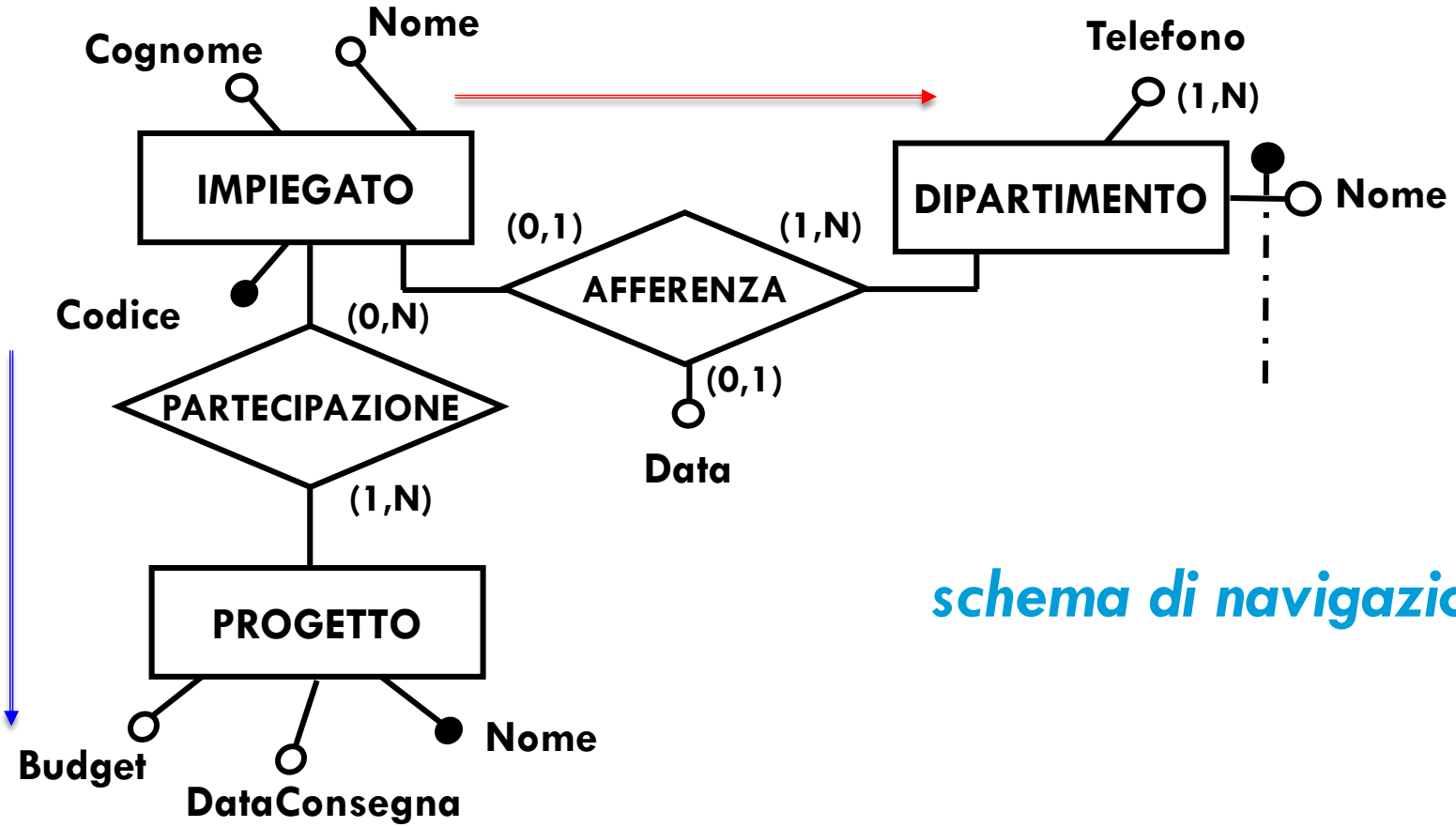
Descrizione delle operazioni

- L'analisi delle operazioni principali richiede la codifica di:
  - tipo dell'operazione: **Interattiva** (I) o **Batch** (B);
  - frequenza: numero medio di esecuzioni in un certo periodo di tempo;
  - schema di navigazione: frammento dello schema E/R interessato dall'operazione sul quale viene evidenziato (con frecce) il "cammino logico" da percorrere per accedere alle informazioni di interesse.
- Per ogni operazione si costruisce una **tavola degli accessi** basata sullo schema di navigazione:
  - il campo costrutto specifica il tipo di concetto (entità o associazione);
  - nel campo accessi si conta il numero degli accessi;
  - il campo tipo è riferito al tipo di operazione: le operazioni di **scrittura** (S) sono più onerose di quelle di **lettura** (L).

Il costo degli accessi in scrittura è in genere considerato doppio rispetto a quello delle letture.

Esempio di valutazione di costo

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



schema di navigazione

Esempio di tavola degli accessi

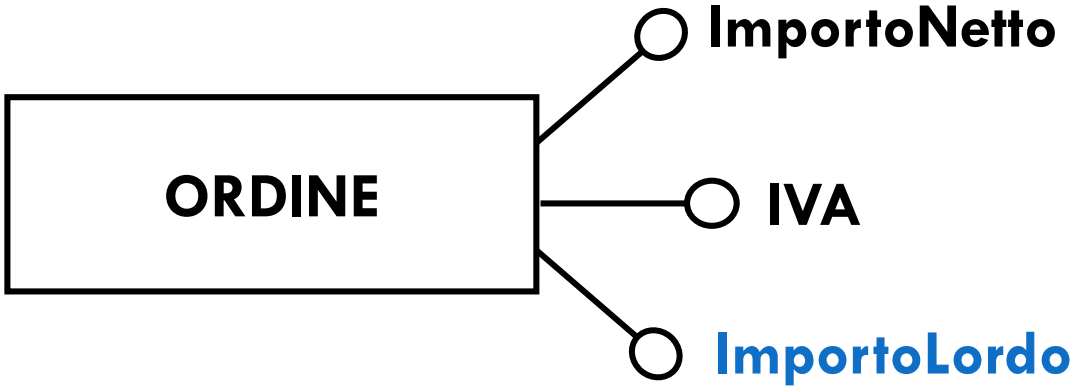
- Per ogni entità e per ogni associazione interessate dall'operazione, la tavola degli accessi riporta il numero di istanze interessate, e il tipo di accesso (L: lettura; S: scrittura)
- Il numero delle istanze si ricava dalla tavola dei volumi mediante semplici operazioni (assumendo uniformità nella distribuzione dei valori): ad esempio in media ogni impiegato partecipa a 6000/2000 = 3 progetti.

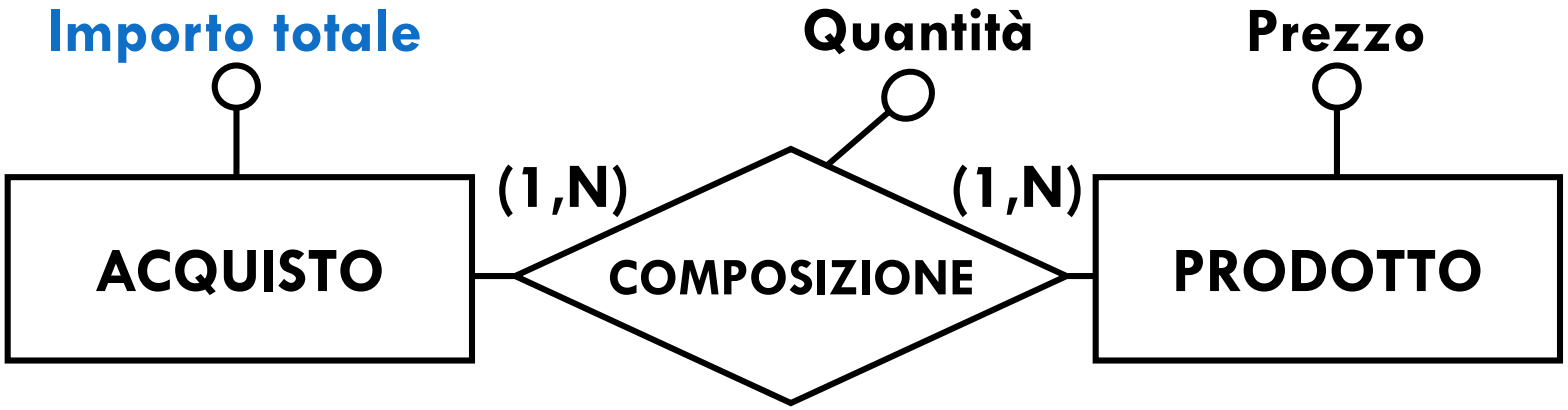
Concetto	Costrutto	Accessi	Tipo
IMPIEGATO	E	1	L
AFFERENZA	A	1	L
DIPARTIMENTO	E	1	L
PARTECIPAZIONE	A	3	L
PROGETTO	E	3	L

Analisi delle ridondanze

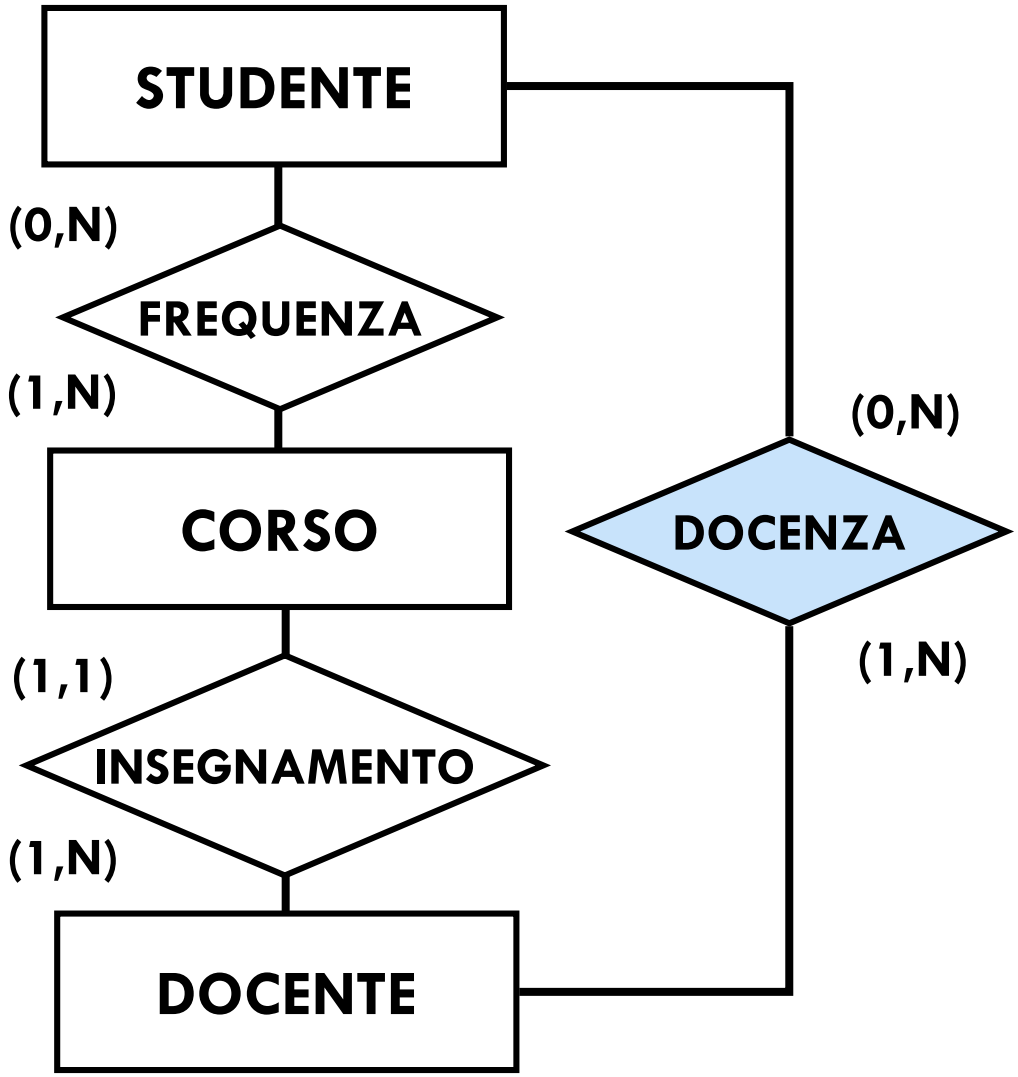
- Una **ridondanza** in uno schema E-R è un'informazione significativa ma derivabile da altre.
- In questa fase si decide se eliminare o meno le ridondanze eventualmente presenti; è quindi comunque importante averle individuate in fase di progettazione concettuale!
- Se si mantiene una ridondanza
  -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

Attributi ridondanti





Associazioni ridondanti



Esempio d'analisi di una ridondanza

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

tabella dei volumi

Concetto	Costrutto	Volume
CITTÀ	E	200
PERSONA	E	1000000
RESIDENZA	A	1000000

Le operazioni…

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

- …e si costruiscono le tavole degli accessi

…in presenza di ridondanza…

Operazione 1

Concetto	Costrutto	Accessi	Tipo
PERSONA	E	1	S
RESIDENZA	A	1	S
CITTÀ	E	1	L
CITTÀ	E	1	S

Aggiornamento = 1L + 1S

Operazione 2

Concetto	Costrutto	Accessi	Tipo
CITTÀ	E	1	L

…in assenza di ridondanza

Operazione 1

Concetto	Costrutto	Accessi	Tipo
PERSONA	E	1	S
RESIDENZA	A	1	S

Operazione 2

Concetto	Costrutto	Accessi	Tipo
CITTÀ	E	1	L
RESIDENZA	A	5000	L

Mantenere o no la ridondanza?

È importante considerare la frequenza delle operazioni:

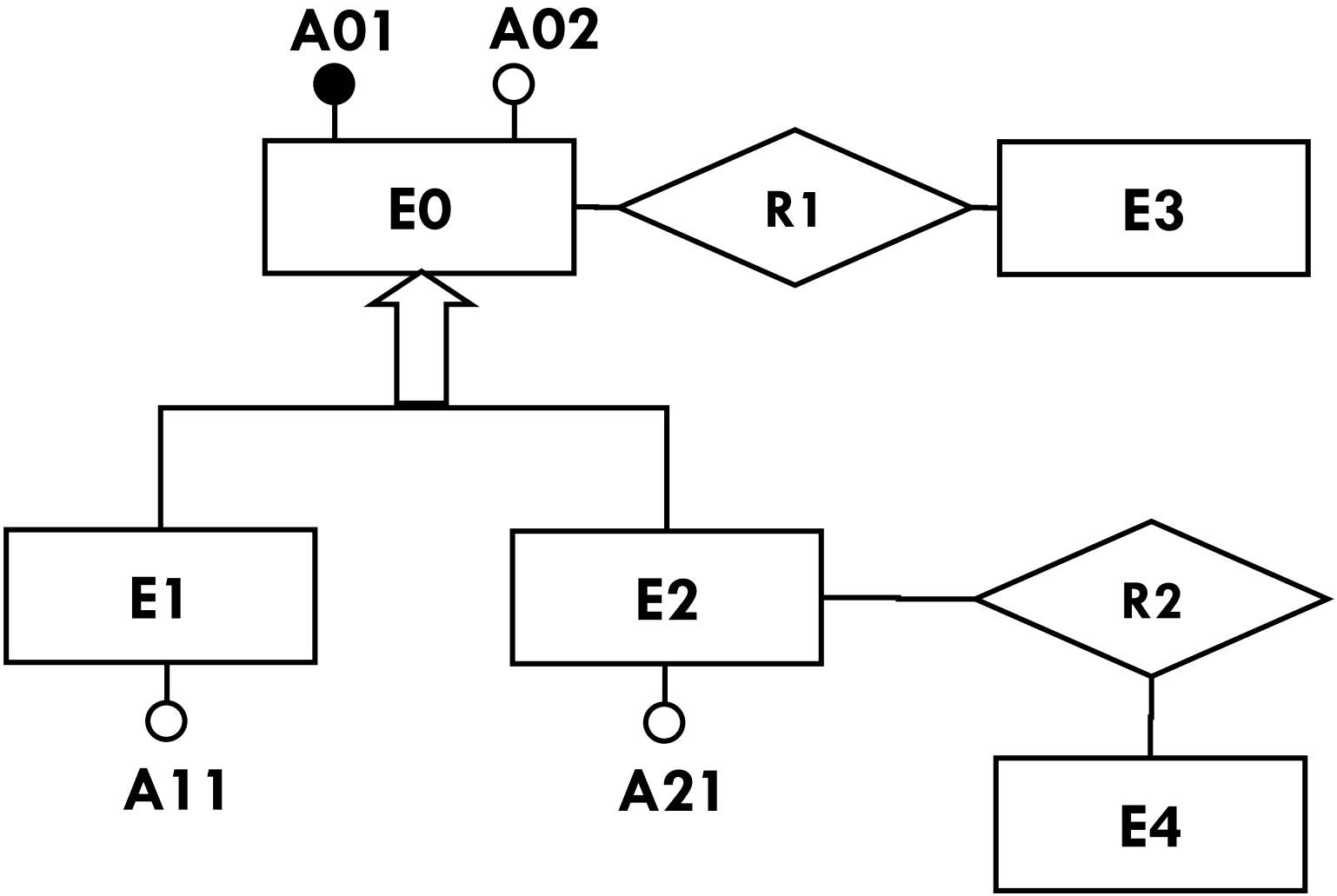
- **con ridondanza**:
  - operazione 1: 1500 accessi in scrittura e 500 accessi in lettura al giorno;
  - operazione 2: 2 accessi in lettura al giorno;
  - totale: 3502 accessi al giorno;
- **senza ridondanza**:
  - operazione 1: 1000 accessi in scrittura al giorno;
  - operazione 2: 10002 accessi in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

Eliminazione delle gerarchie

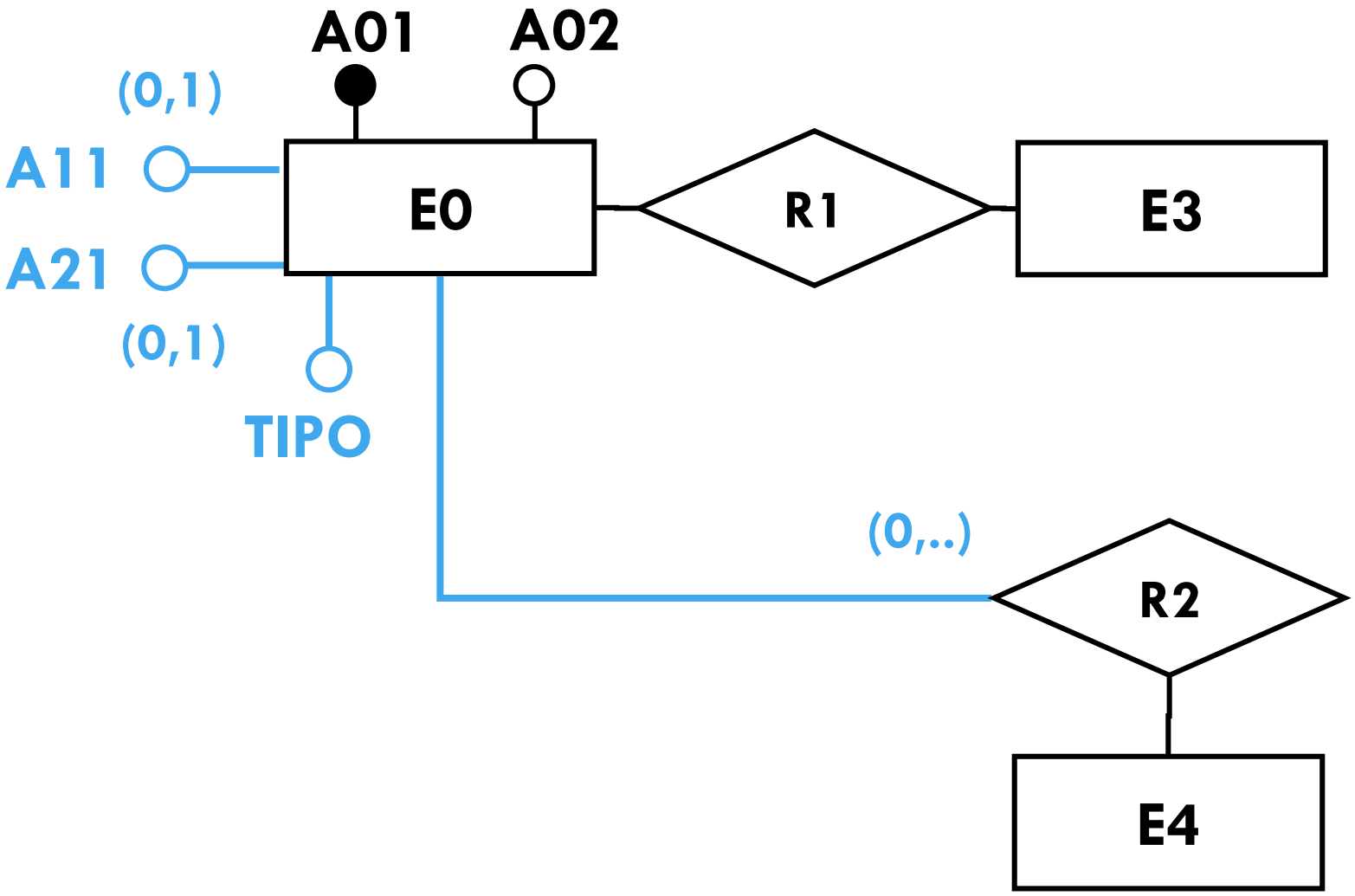
- Il modello relazionale non può rappresentare direttamente le gerarchie di generalizzazione.
- Entità e associazioni sono invece direttamente rappresentabili.
- Si eliminano perciò le gerarchie, sostituendole con entità e relazioni.

- Vi sono 3 possibilità (più altre soluzioni intermedie):
  - accorpare le entità figlie nel genitore (collasso verso l'alto);
  - accorpare il genitore nelle entità figlie (collasso verso il basso);
  - sostituire la generalizzazione con associazioni.

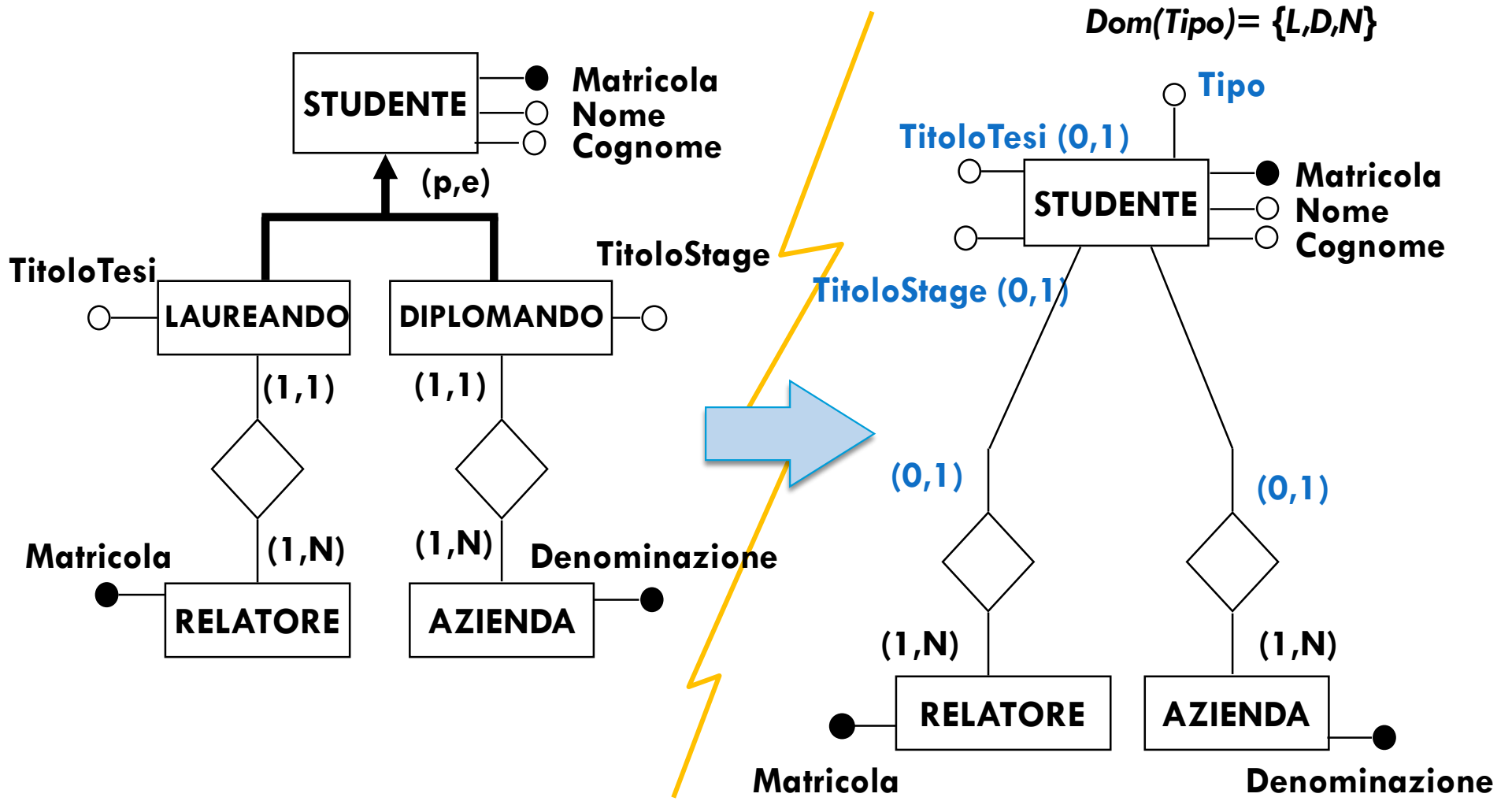
Schema di riferimento



1. Collasso verso l'alto…

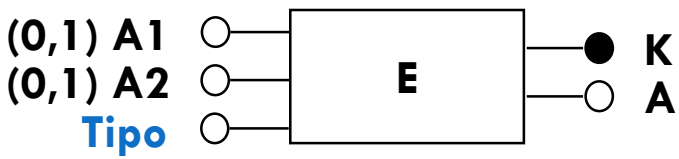


Esempio


STUDENTE
Matricola
Nome
Cognome
(p,e)
TitoloTesi
TitoloStage
LAUREANDO
DIPLOMANDO
(1,1)
(1,1)
Matricola
(1,N)
(1,N)
Denominazione
RELATORE
AZIENDA
Dom(Tipo)= {L,D,N}
Tipo
TitoloTesi (0,1)
STUDENTE
Matricola
Nome
Cognome
TitoloStage (0,1)
(0,1)
(0,1)
(1,N)
(1,N)
RELATORE
AZIENDA
Matricola
Denominazione

Collasso verso l'alto: osservazioni

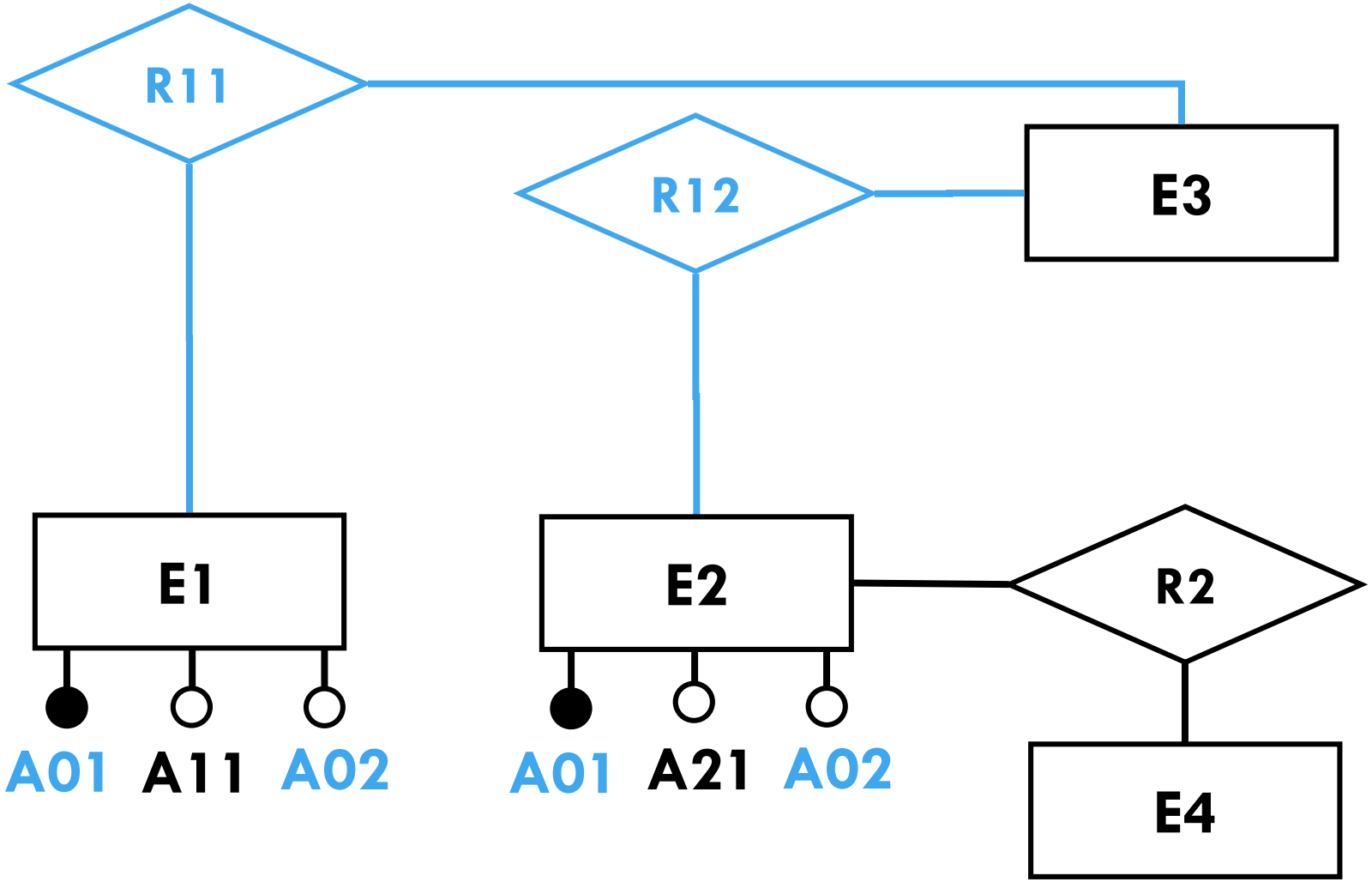
"Tipo" è un attributo *selettore* che specifica se una singola istanza di E appartiene a una delle N sotto-entità.



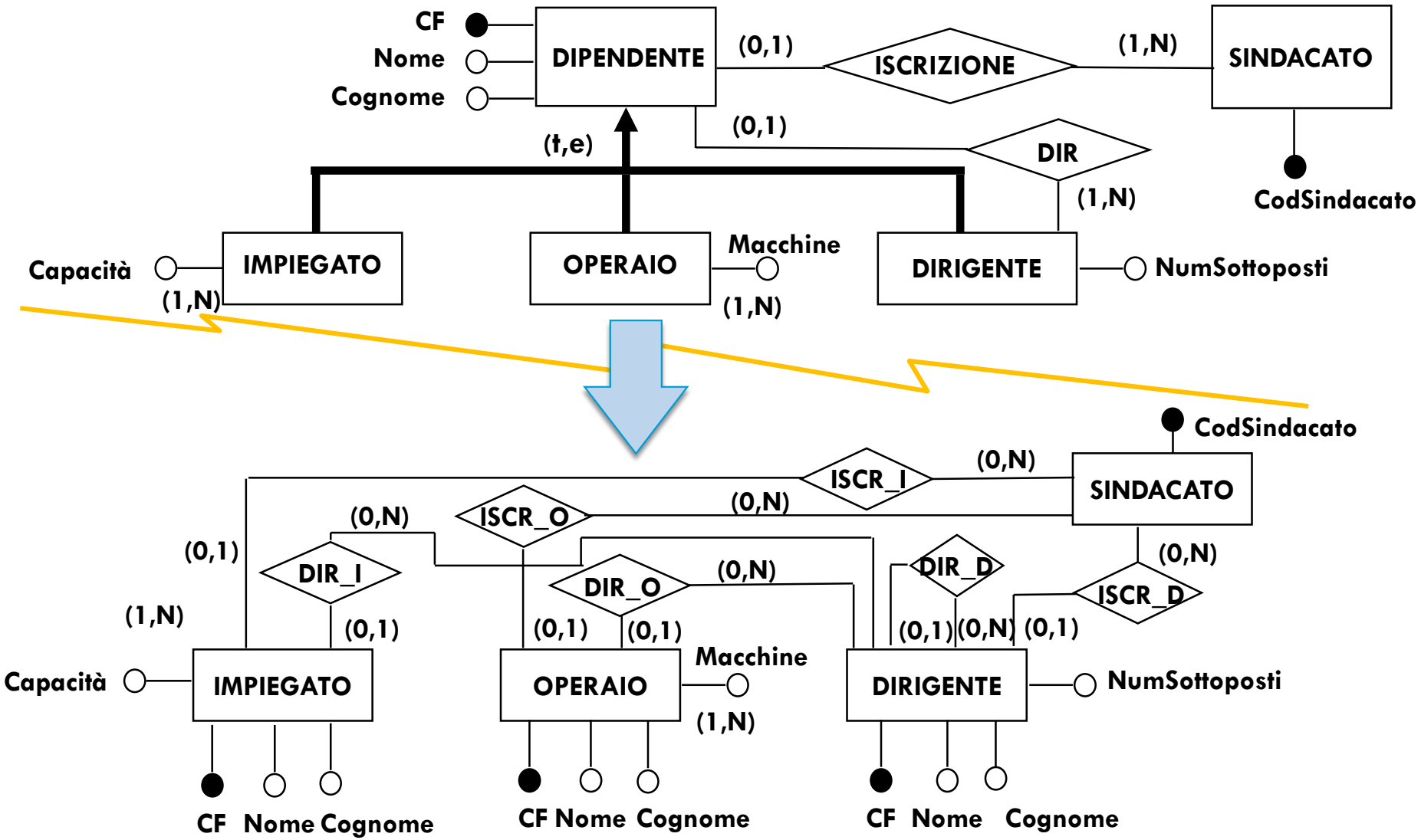
- **Copertura**
  - totale esclusiva: Tipo assume N valori, quante sono le sotto-entità;
  - parziale esclusiva: Tipo assume N+1 valori; il valore in più serve per le istanze che non appartengono a nessuna sotto-entità;
  - sovrapposta: occorrono tanti selettori quante sono le sotto-entità, ciascuno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

- Le eventuali associazioni connesse alle sotto-entità si trasportano su E, le eventuali cardinalità minime diventano 0.

2. Collasso verso il basso…

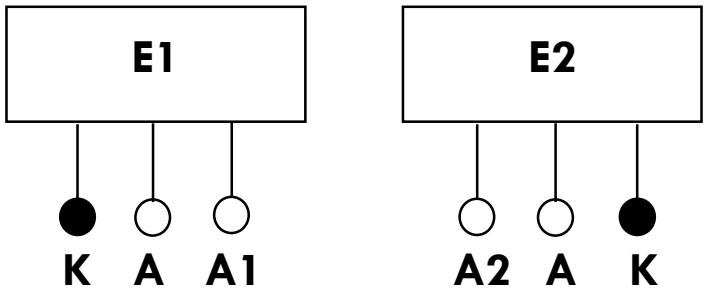


2. Esempio

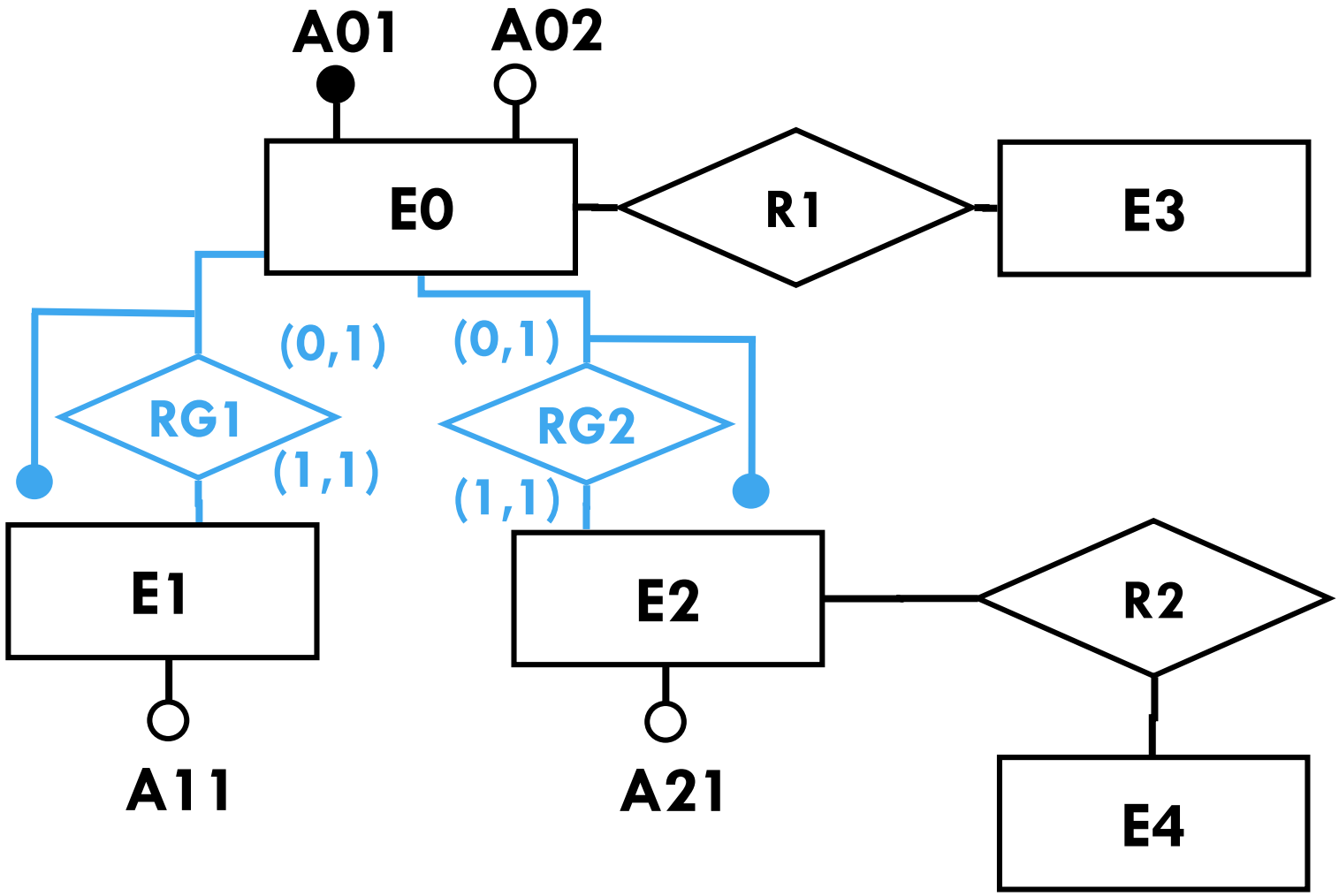

CF
Nome
Cognome
DIPENDENTE
(0,1)
ISCRIZIONE
(1,N)
SINDACATO
CodSindacato
(t,e)
(0,1)
DIR
(1,N)
Capacità
IMPIEGATO
(1,N)
OPERAIO
Macchine
(1,N)
DIRIGENTE
NumSottoposti
CodSindacato
ISCR_I
(0,N)
SINDACATO
(0,N)
ISCR_O
(0,N)
(0,1)
DIR_I
DIR_O
(0,N)
DIR_D
(0,N)
ISCR_D
(1,N)
(0,1)
(0,1)
(0,1)
(0,1)
(0,N)
(0,1)
Capacità
IMPIEGATO
OPERAIO
Macchine
(1,N)
DIRIGENTE
NumSottoposti
CF Nome Cognome
CF Nome Cognome
CF Nome Cognome

Collasso verso il basso: osservazioni

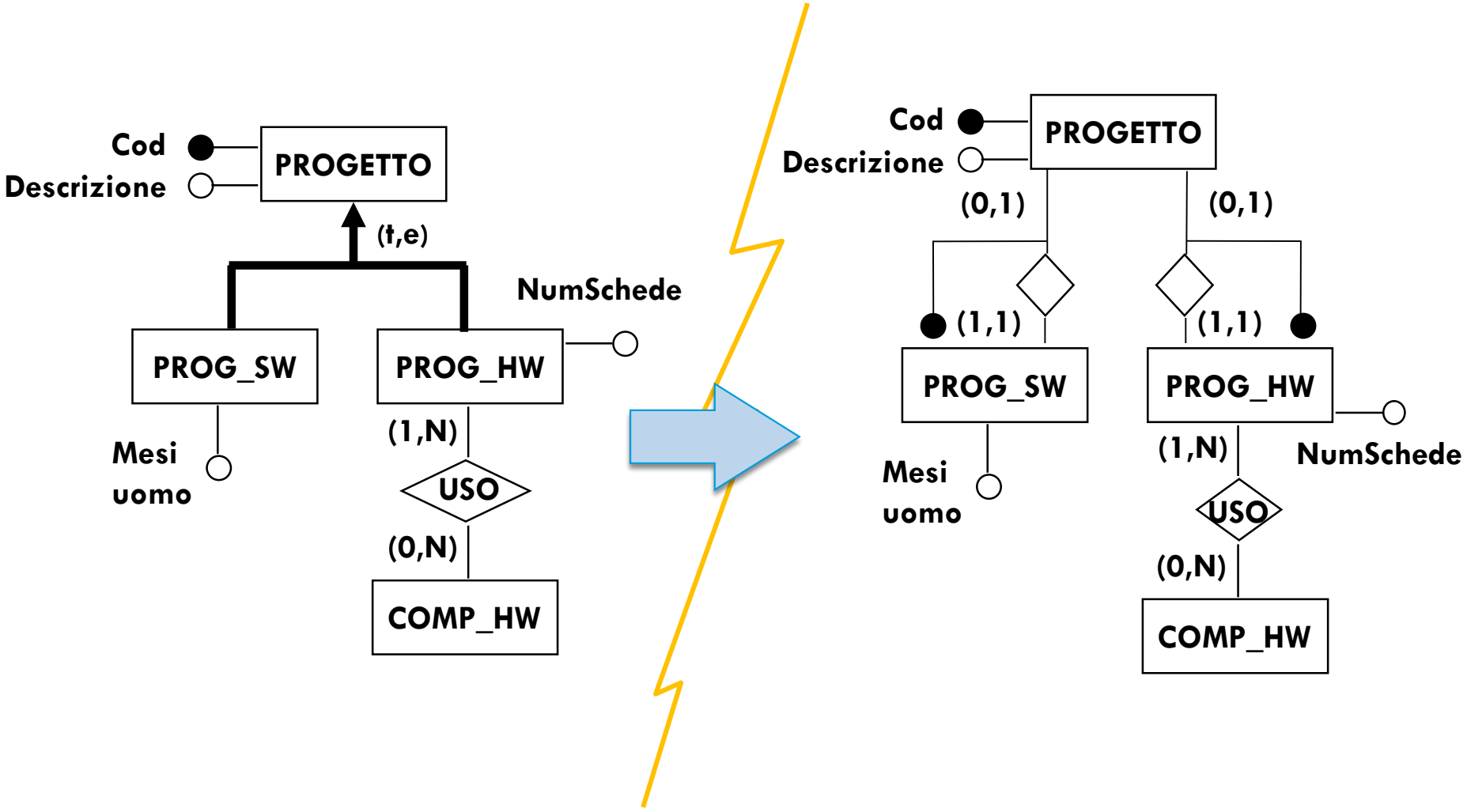
- Se la copertura non è completa il collasso verso il basso non si può applicare:
  - non si saprebbe infatti dove collocare le istanze di E che non sono né in E1, né in E2.
- Se la copertura non è esclusiva introduce ridondanza:
  - una certa istanza può essere sia in E1 sia in E2, e quindi si rappresentano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



3. Sostituire con associazioni…



3. Esempio


Cod
Descrizione
PROGETTO
(t,e)
NumSchede
PROG_SW
PROG_HW
Mesi uomo
(1,N)
USO
(0,N)
COMP_HW
Cod
Descrizione
PROGETTO
(0,1)
(0,1)
(1,1)
(1,1)
PROG_SW
PROG_HW
Mesi uomo
(1,N)
NumSchede
USO
(0,N)
COMP_HW

Sostituire con associazioni: osservazioni

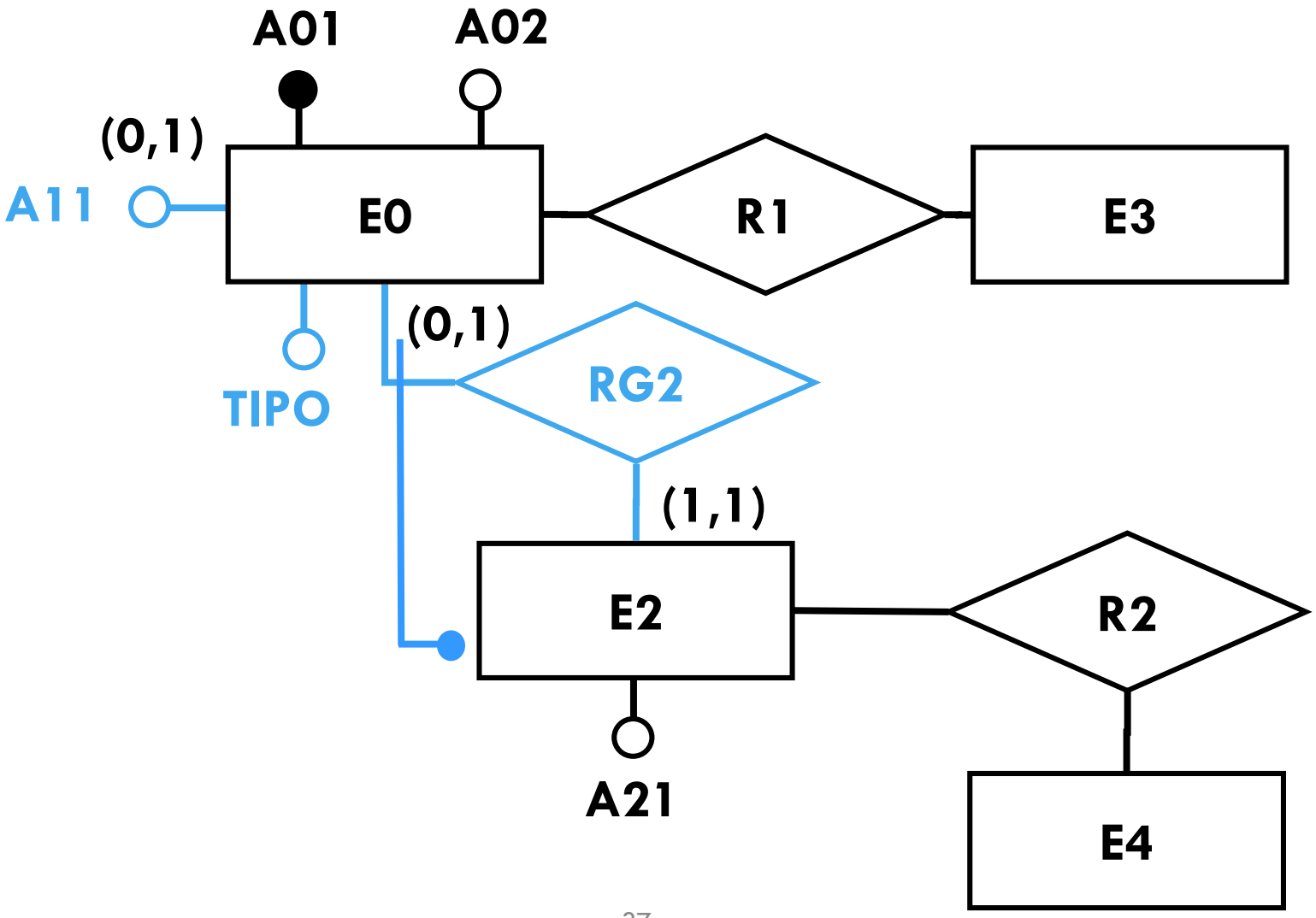
- Tutte le entità vengono mantenute: le entità figlie sono in associazione binaria con l'entità padre e sono identificate esternamente.

- La sostituzione con associazioni è sempre possibile indipendentemente dalla copertura della gerarchia.

Quale alternativa scegliere?

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

Una soluzione ibrida…

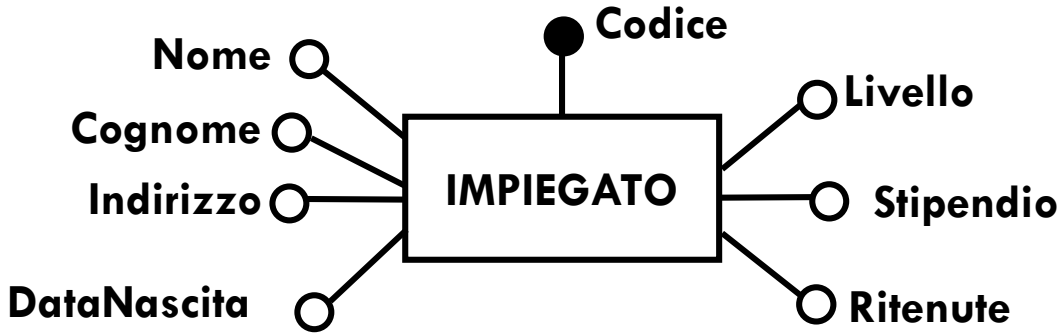


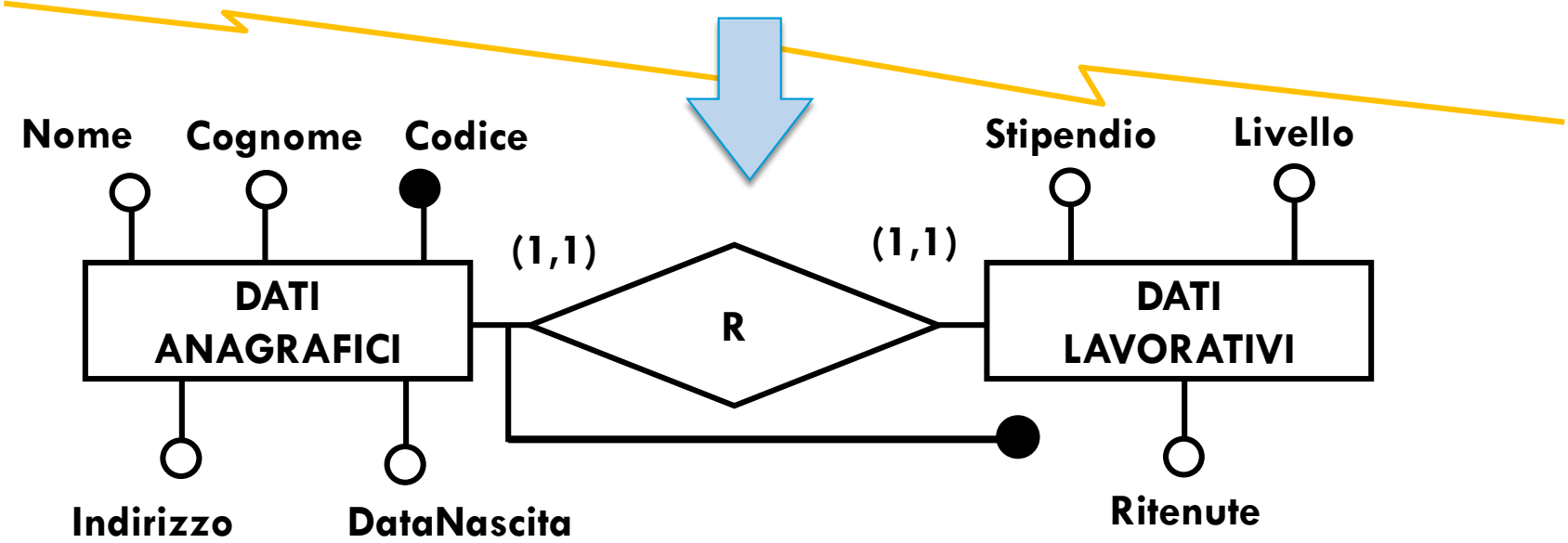
Partizionamenti e accorpamenti

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

Partizionamento verticale di entità

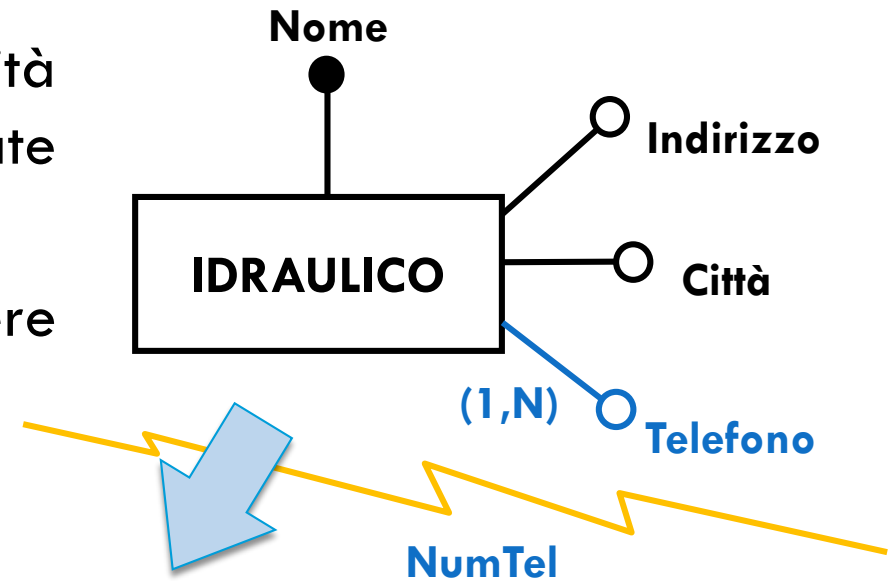
- Si separano gli attributi in gruppi omogenei:

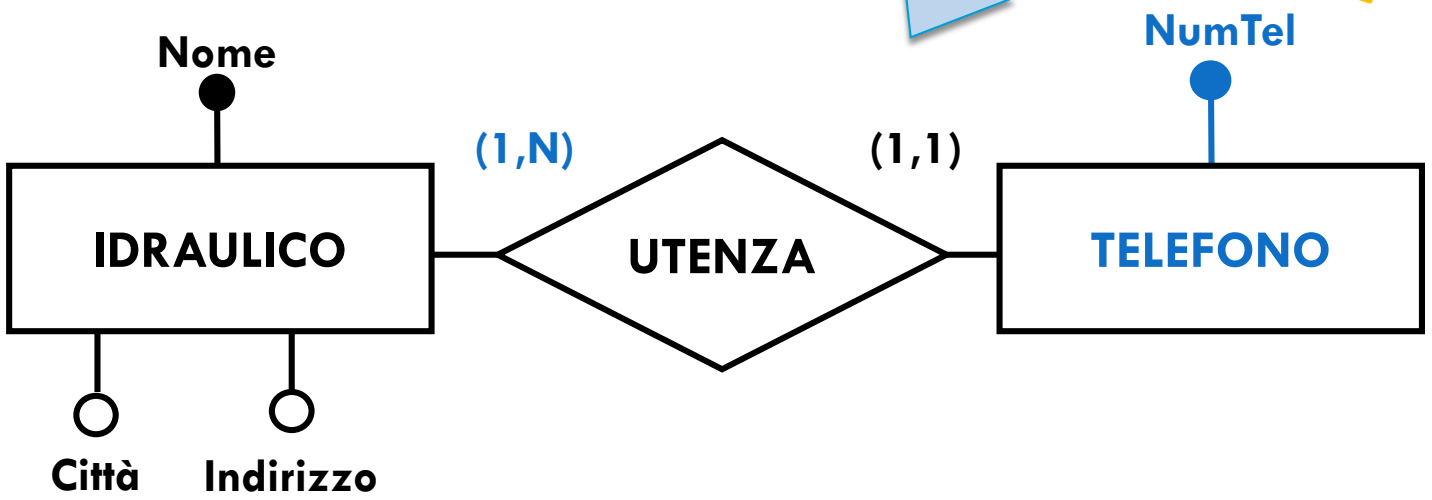




Eliminazione di attributi multivalore (1)

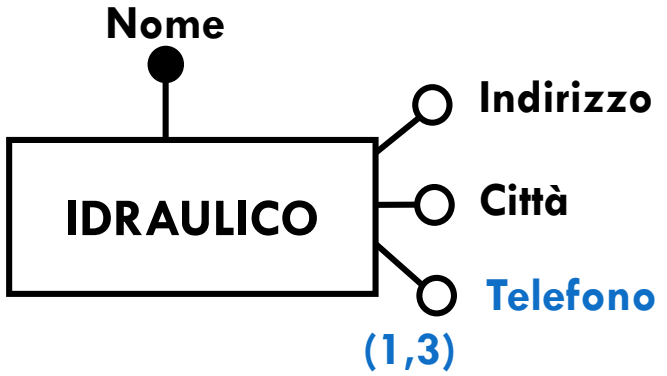
- Si introduce una nuova entità le cui istanze sono identificate dai valori dell'attributo.
- L'associazione può essere uno a molti o molti a molti.

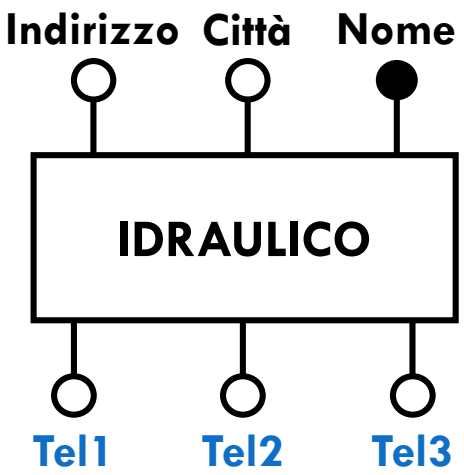




Eliminazione di attributi multivalore (2)

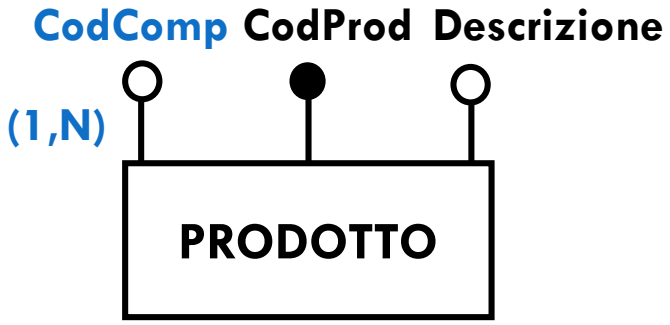
- Se è nota la cardinalità massima K di un attributo multivalore allora è possibile prevedere K attributi a singolo valore.

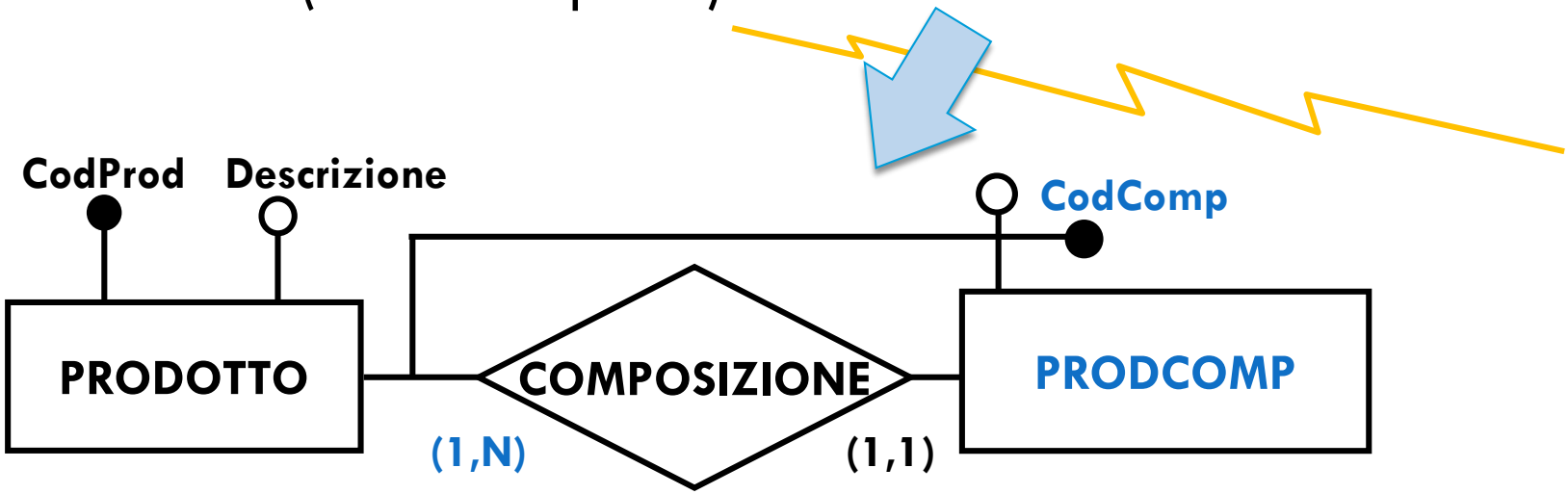




Eliminazione di attributi multivalore (3)

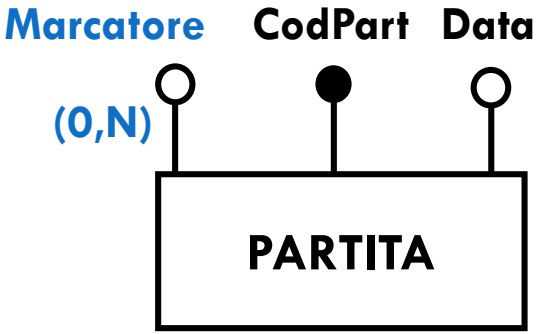
- Se un valore dell'attributo multivalore compare una sola volta nella ripetizione esso può costituire l'identificatore della nuova entità (o una sua parte).

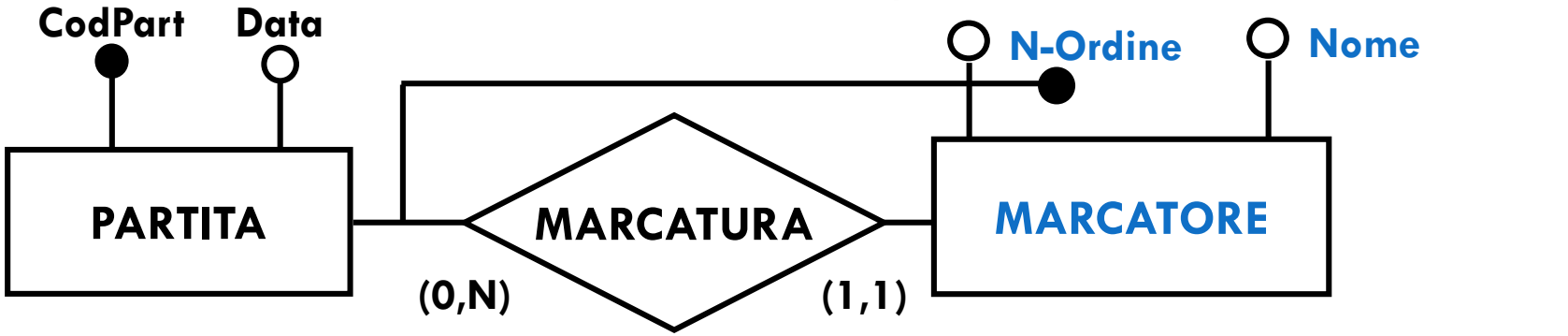




Eliminazione di attributi multivalore (4)

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





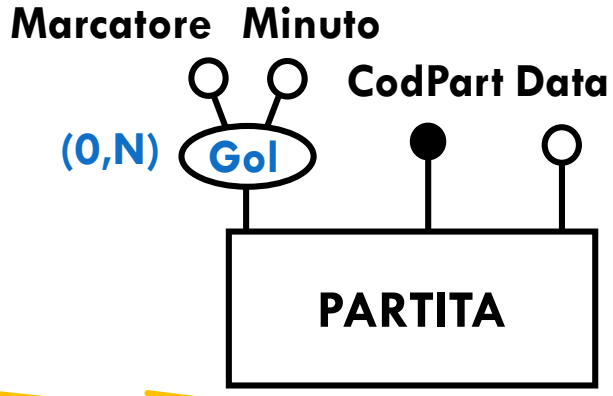
Attributi multivalore: note (1)

- Per affinità con la concezione insiemistica che sta alla base del modello E/R sarebbe naturale ipotizzare che i valori di un attributo multivalore siano tutti distinti.

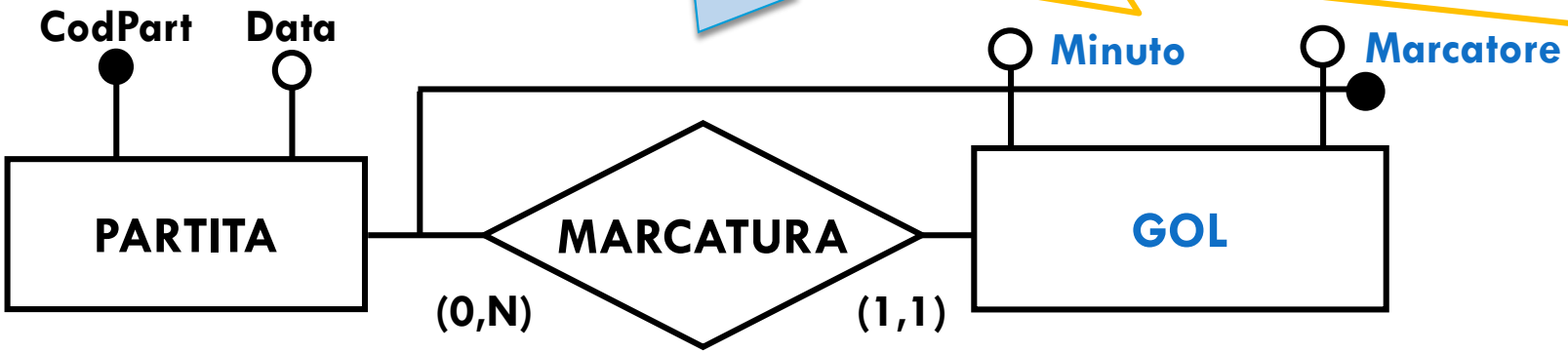
- Alcuni autori e tool di progettazione (es. DB-Main) fanno riferimento a un'estensione del modello in cui è ammessa la duplicazione di valori. In quest'ottica assumono significato gli esempi dei due lucidi precedenti.

- Attenzione! In generale la presenza di valori duplicati in un attributo multivalore può indicare una progettazione errata e/o incompleta. Es. Relativamente ai marcatori della partita ci interessa memorizzare solo il nome? Non è preferibile modellare il concetto di gol, riportando anche il minuto in cui è stato realizzato?

Attributi multivalore: note (2)

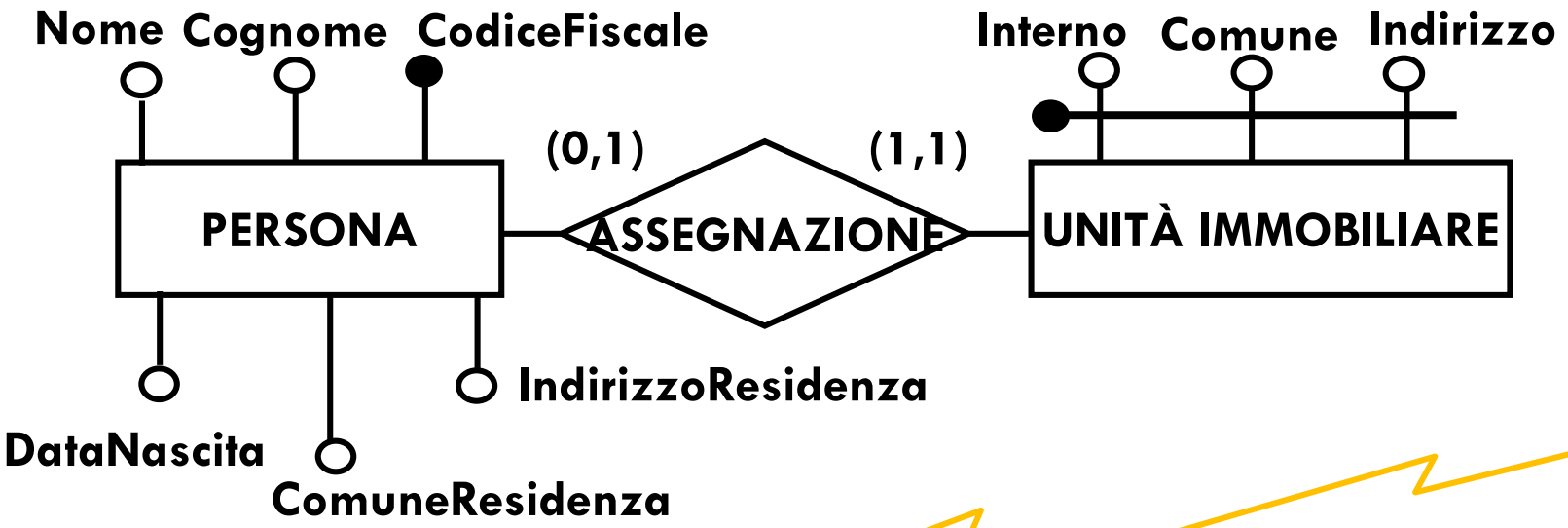


Questa modellazione è più completa e in questo contesto la ripetizione di valori perde il suo significato.

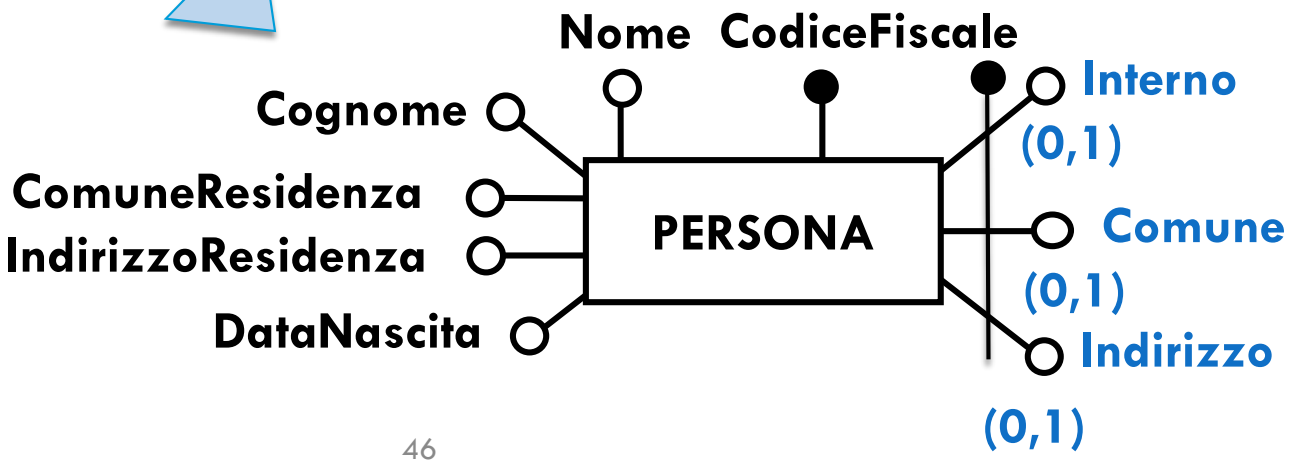


Attenzione! Questo schema ammette che possano essere fatti due gol nello stesso minuto, purché da giocatori diversi…

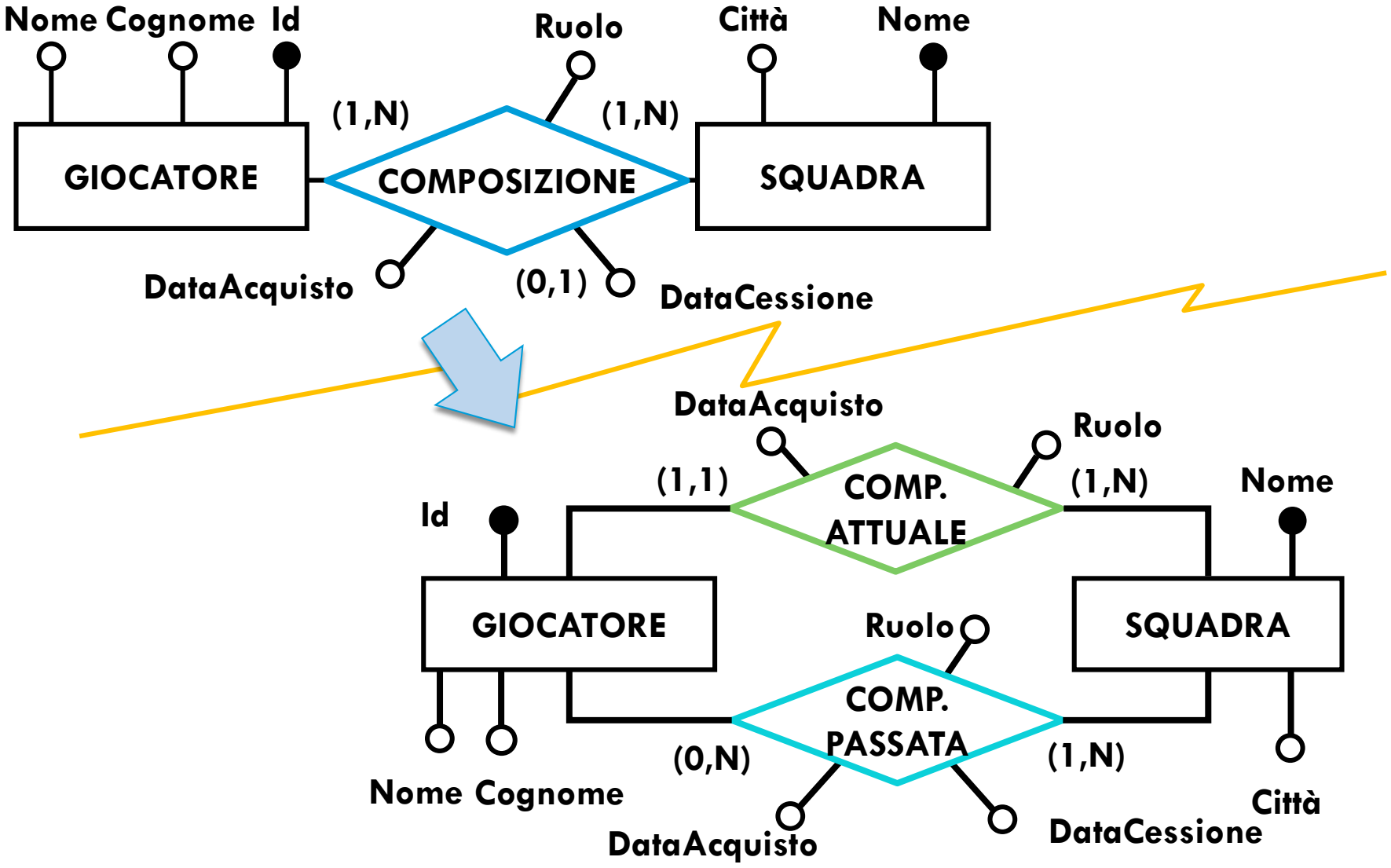
Accorpamento di entità



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



Partizionamento orizzontale di associazioni

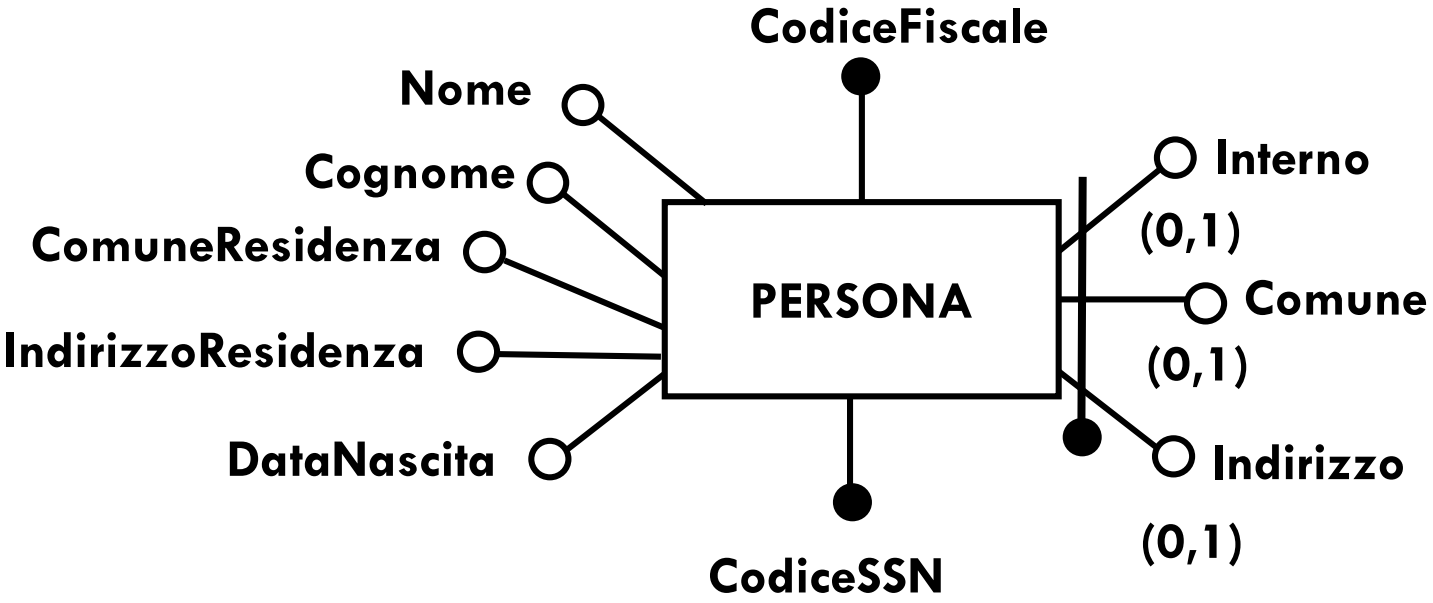


Scelta degli identificatori principali

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

Identificatori principali: esempio

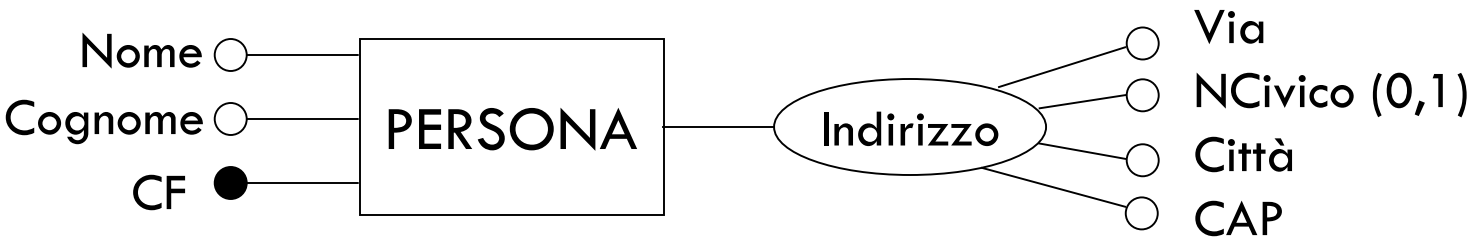
- L'identificatore {Interno, Comune, Indirizzo} è opzionale, quindi non può essere scelto come chiave primaria.
- Tra gli attributi CodiceFiscale e CodiceSSN la scelta dipende da quale fra questi è più frequentemente usato per accedere ai dati di una persona.



Traduzione delle entità

Idea di base:

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



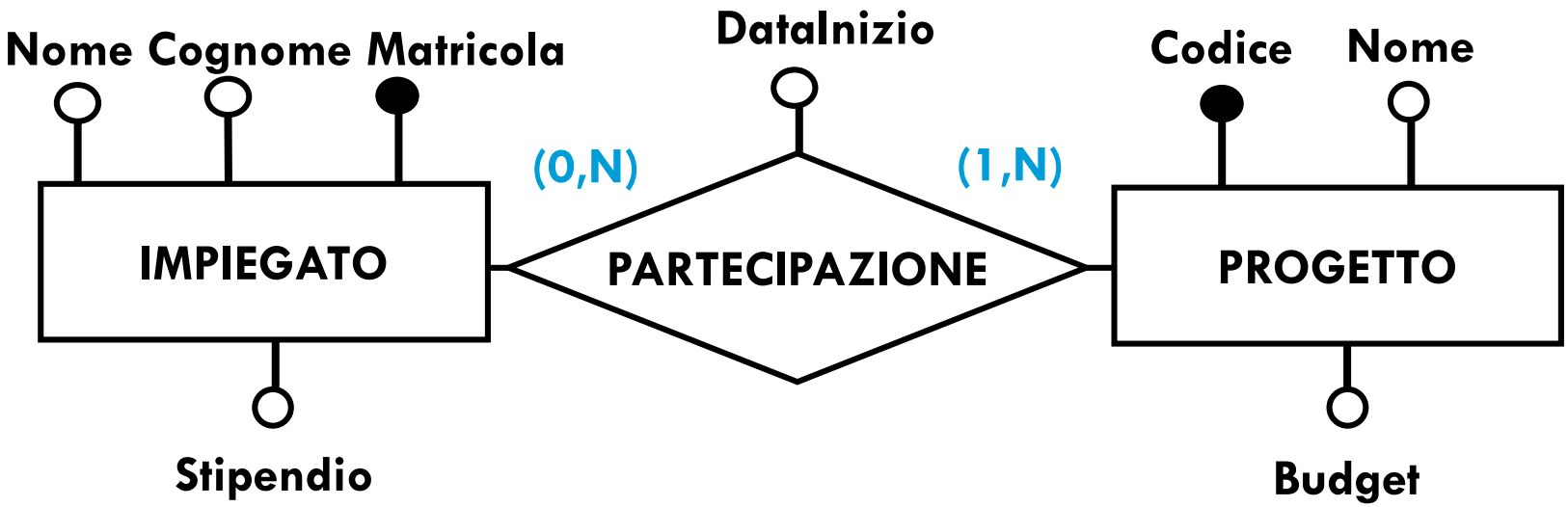
PERSONE(CF, Cognome, Nome, Via, NCivico*, Città, CAP)

Traduzione delle associazioni

Idea di base:

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

Entità e associazione molti a molti



IMPIEGATI(Matricola, Nome, Cognome, Stipendio)

PROGETTI(Codice, Nome, Budget)

PARTECIPAZIONI(Matricola, Codice, DataInizio)

FK: Matricola REFERENCES Impiegati

FK: Codice REFERENCES Progetti

Nomi delle foreign key: ridenominazione

- Non è ovviamente necessario mantenere, per gli attributi chiave della relazione che traduce l'associazione, gli stessi nomi delle primary key referenziate, conviene piuttosto far ricorso a nomi più espressivi.

PARTECIPAZIONI(Impiegato, CodProgetto, DataInizio)

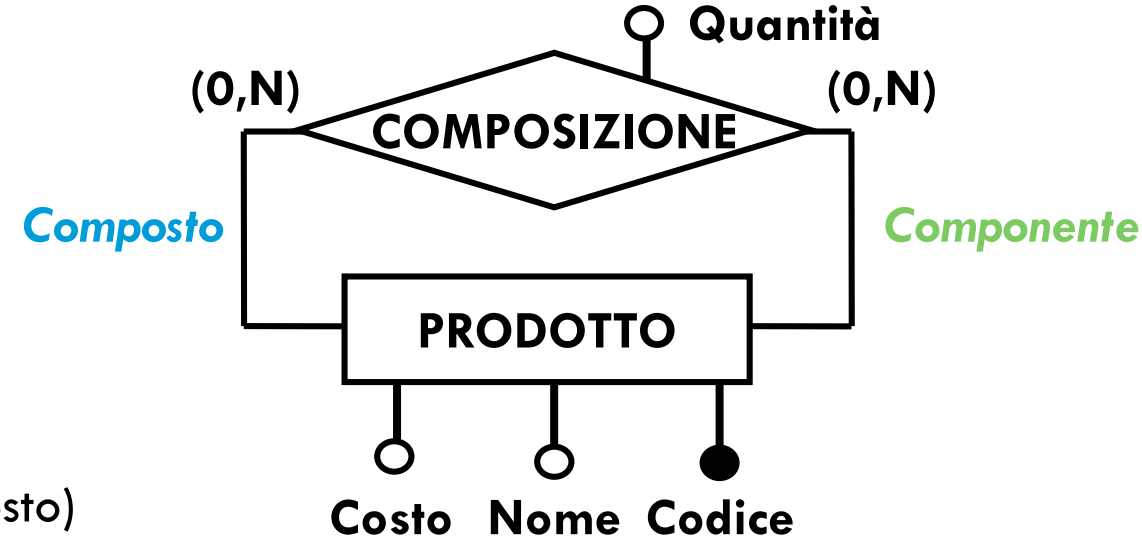
FK: Impiegato REFERENCES Impiegati

FK: CodProgetto REFERENCES Progetti

- Ovviamente se le entità collegate hanno un attributo con lo stesso nome la ridenominazione è obbligatoria!

Associazioni ad anello molti a molti

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



PRODOTTI(Codice, Nome, Costo)

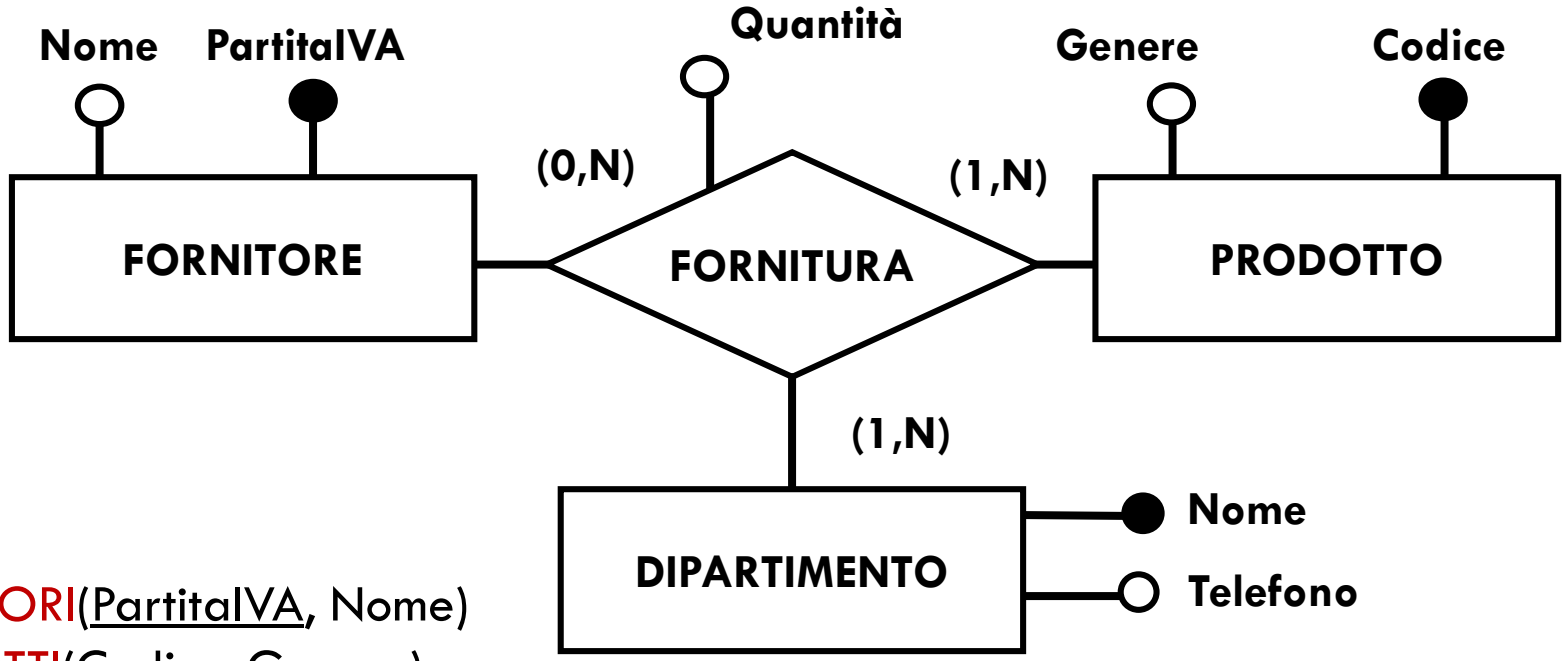
COMPOSIZIONI(Composto, Componente, Quantità)

FK: Composto REFERENCES Prodotti

FK: Componente REFERENCES Prodotti

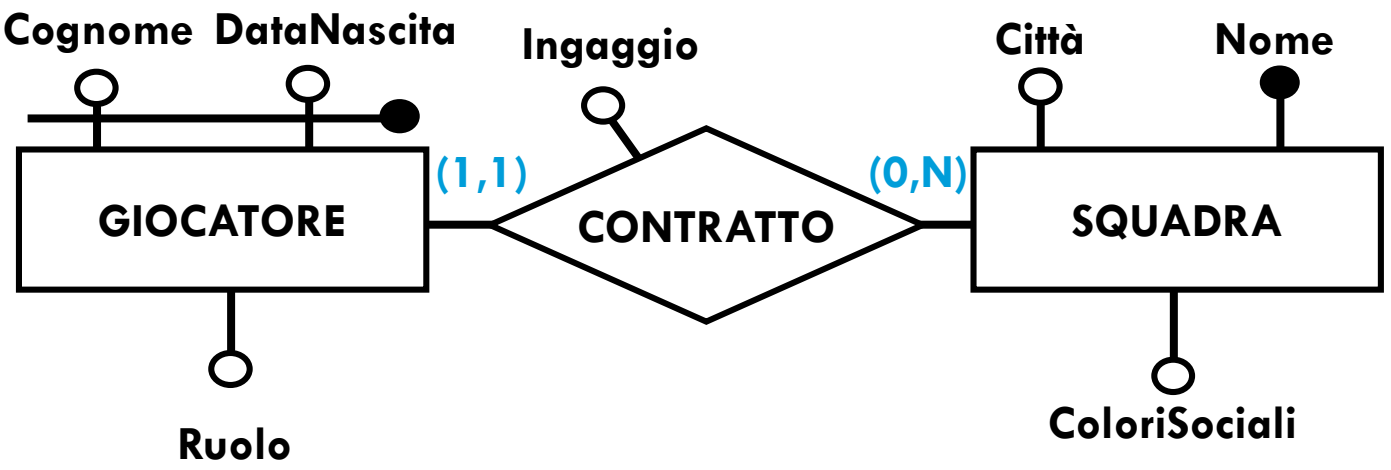
Associazioni n-arie molti a molti

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



FORNITORI(PartitaIVA, Nome)
PRODOTTI(Codice, Genere)
DIPARTIMENTI(Nome, Telefono)
FORNITURE(Fornitore, Prodotto, Dipartimento, Quantità)
FK: …

Associazioni uno a molti (1)



GIOCATORI(Cognome, DataNascita, Ruolo)

SQUADRE(Nome, Città, ColoriSociali)

CONTRATTI(CognomeGiocatore, DataNascitaGiocatore, Squadra, Ingaggio)

FK: (CognomeGiocatore, DataNascitaGiocatore) REFERENCES Giocatori

FK: Squadra REFERENCES Squadre

Il Nome della Squadra non fa parte della chiave di Contratto *(perché?)*

Associazioni uno a molti (2)

- Poiché un giocatore ha un contratto con una sola squadra, nella relazione Contratto un giocatore non può apparire in più tuple.

- Si può pertanto adottare anche una soluzione più compatta, che fa uso di 2 sole relazioni:

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

  FK: Squadra REFERENCES Squadre

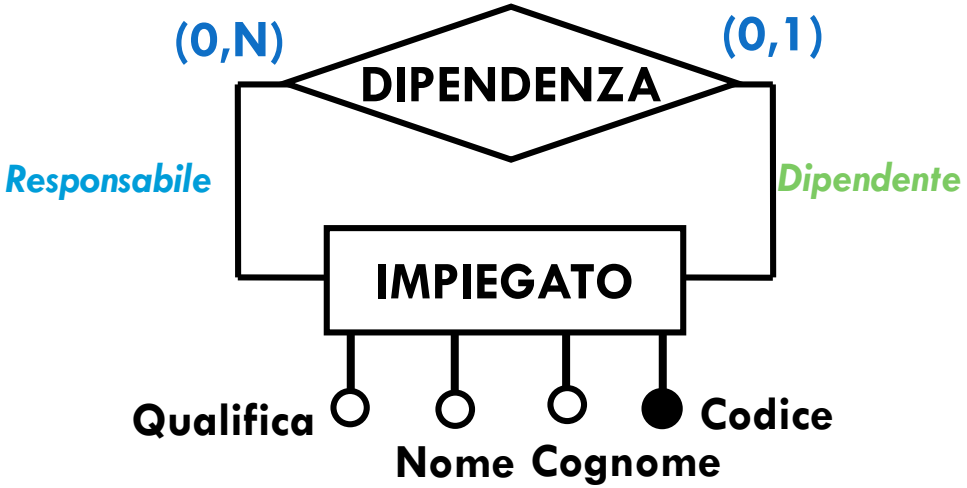
  SQUADRE(Nome, Città, ColoriSociali)

  che corrisponde a tradurre l'associazione 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).

Associazioni ad anello uno a molti

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



1 relazione:

IMPIEGATI(Codice, Nome, Cognome, Qualifica, Responsabile*)

FK: Responsabile REFERENCES Impiegati

2 relazioni:

IMPIEGATI(Codice, Nome, Cognome, Qualifica)

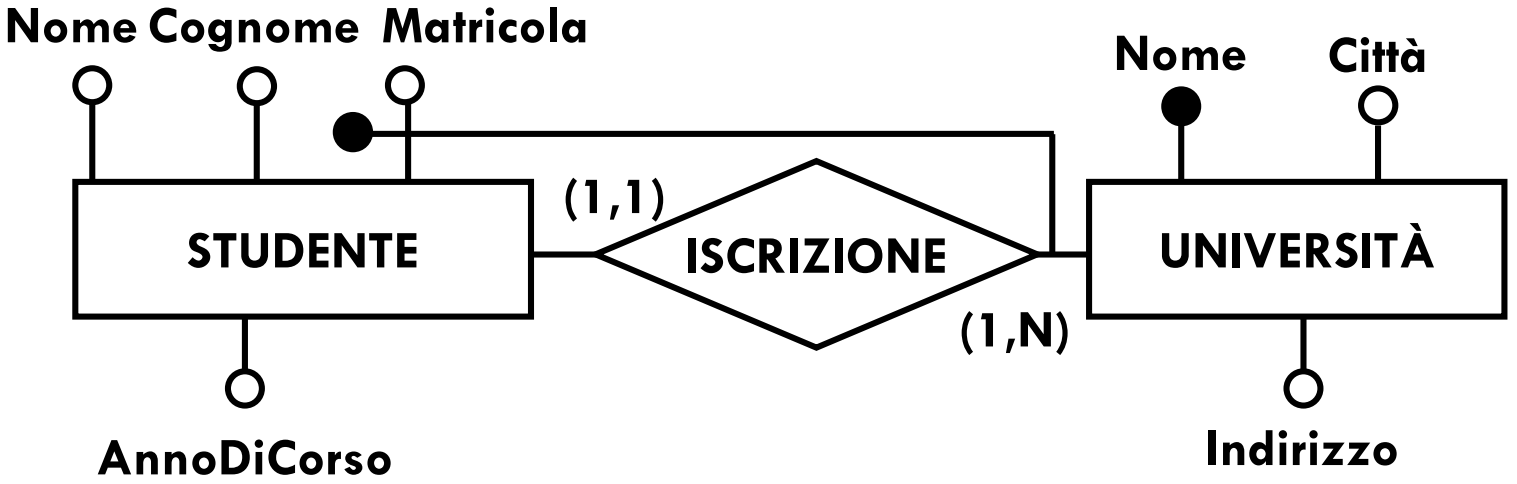
DIPENDENZE(Dipendente, Responsabile)

FK: Dipendente REFERENCES Impiegati

FK: Responsabile REFERENCES Impiegati

Entità con identificazione esterna

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



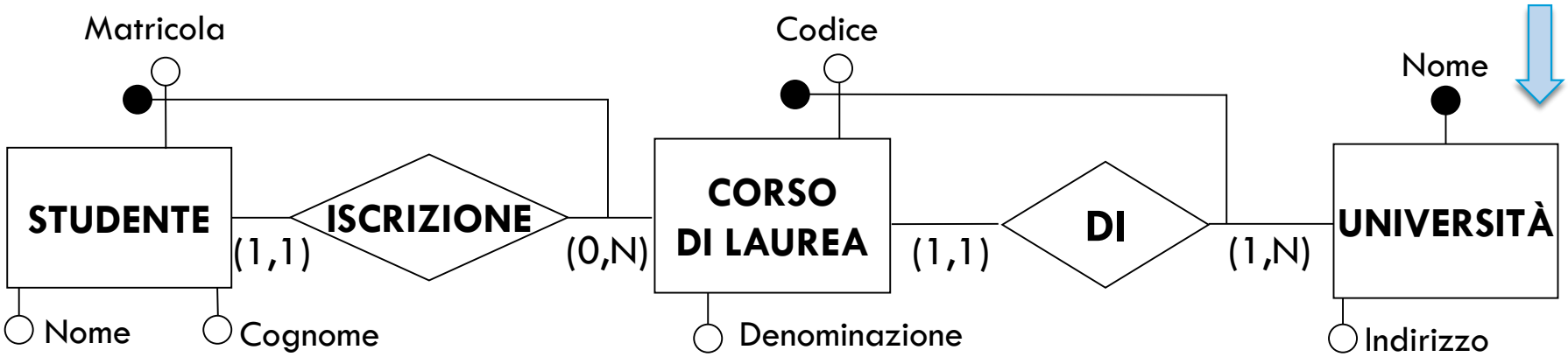
STUDENTI(Matricola, Università, Cognome, Nome, AnnoDiCorso)

FK: Università REFERENCES Università

UNIVERSITÀ(Nome, Città, Indirizzo)

Identificazioni esterne: una precisazione

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



UNIVERSITÀ(Nome, Indirizzo)

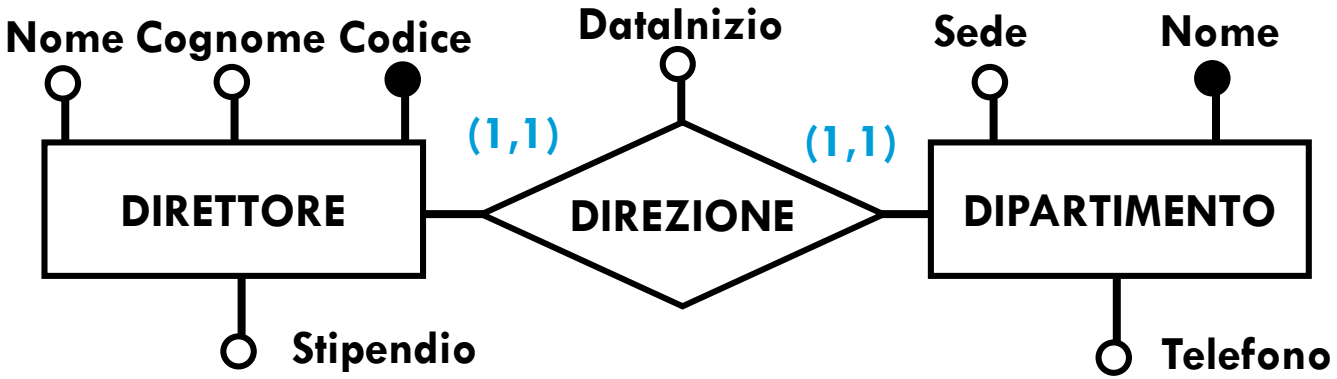
CORSIDILAUREA(Università, Codice, Denominazione)

STUDENTI(Università, CodiceCdL, Matricola, Cognome, Nome)

FK:....

Associazioni uno a uno (1)

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



Tre relazioni:

DIRETTORI(Codice, Nome, Cognome, Stipendio)

DIPARTIMENTI(Nome, Sede, Telefono)

DIREZIONI(Direttore, Dipartimento, DataInizio)

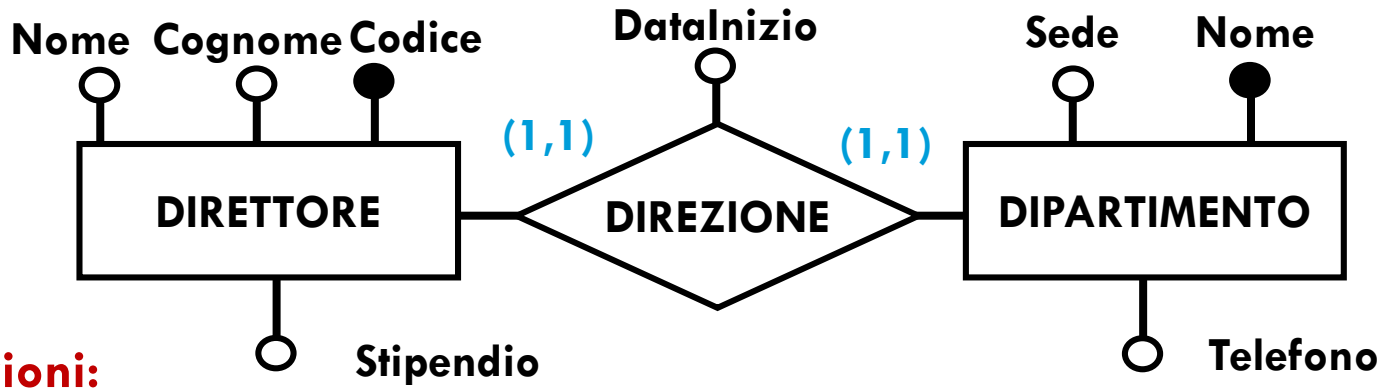
FK:…

Unique(Dipartimento)

L'identificatore di una delle due entità è scelto come chiave primaria, l'altro dà origine a una chiave alternativa.

La scelta dipende dall'importanza relativa delle chiavi.

Associazioni uno a uno (2)



Due relazioni:

DIRETTORI(Codice, Nome, Cognome, Stipendio, Dipartimento, DataInizio)

FK: Dipartimento REFERENCES Dipartimenti

Unique(Dipartimento)

DIPARTIMENTI(Nome, Sede, Telefono)

oppure

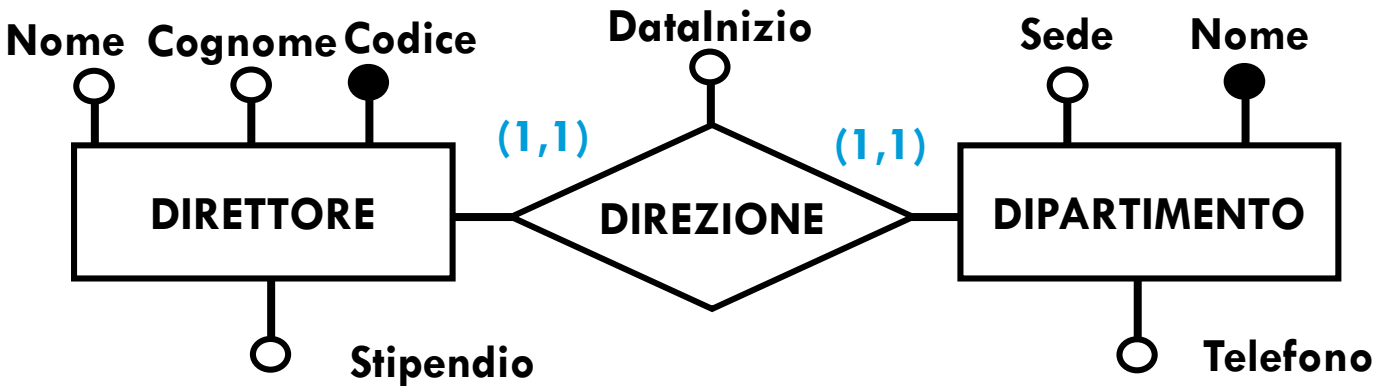
DIRETTORI(Codice, Nome, Cognome, Stipendio)

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

FK: Direttore REFERENCES Direttori

Unique(Direttore)

Associazioni uno a uno (3)



Una relazione:

DIRETTORI(<u>Codice</u>, Nome, Cognome, Stipendio, DataInizio, Dipartimento, Sede, Telefono)

Unique(Dipartimento)

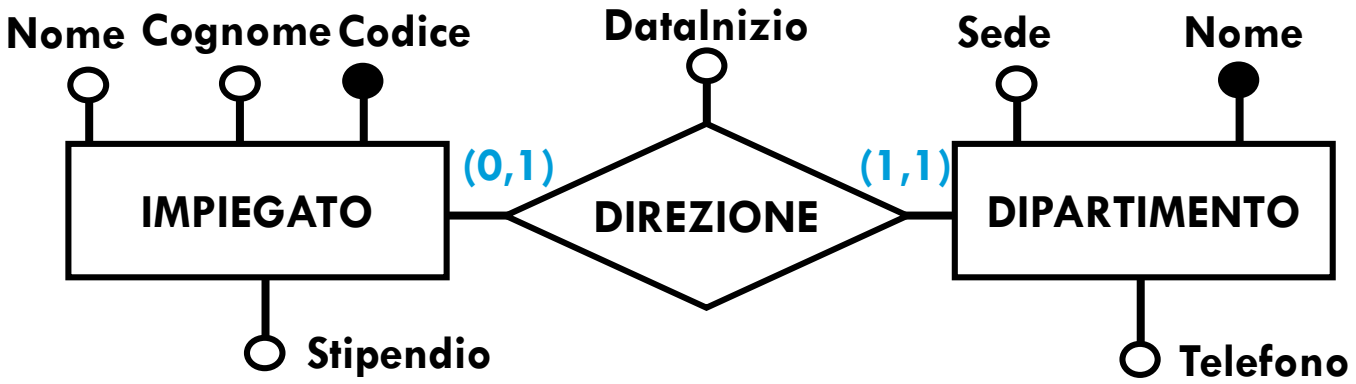
<u>oppure</u>

DIPARTIMENTI(<u>Nome</u>, Sede, Telefono, Direttore, NomeDirettore, CognomeDirettore, Stipendio, DataInizio)

Unique(Direttore)

Associazioni uno a uno con opzionalità (1)

- Se min-card(E,R)=0, tradurre l'associazione R inglobandola in E non è in generale una buona scelta (dipende dai volumi dei dati in gioco).



IMPIEGATI(Codice, Nome, Cognome, Stipendio, Dipartimento*, DataInizio*)

FK: Dipartimento REFERENCES Dipartimenti

Unique(Dipartimento)

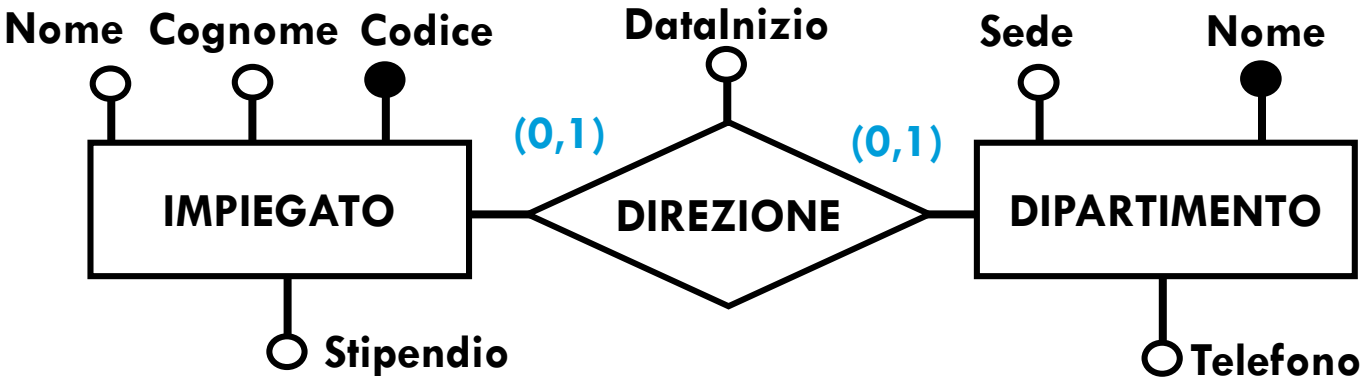
CHECK (((Dipartimento IS NOT NULL) AND (DataInizio IS NOT NULL)) OR
((Dipartimento IS NULL) AND (DataInizio IS NULL)))

TROPPI VALORI NULLI!

DIPARTIMENTI(Nome, Sede, Telefono)

Associazioni uno a uno con opzionalità (2)

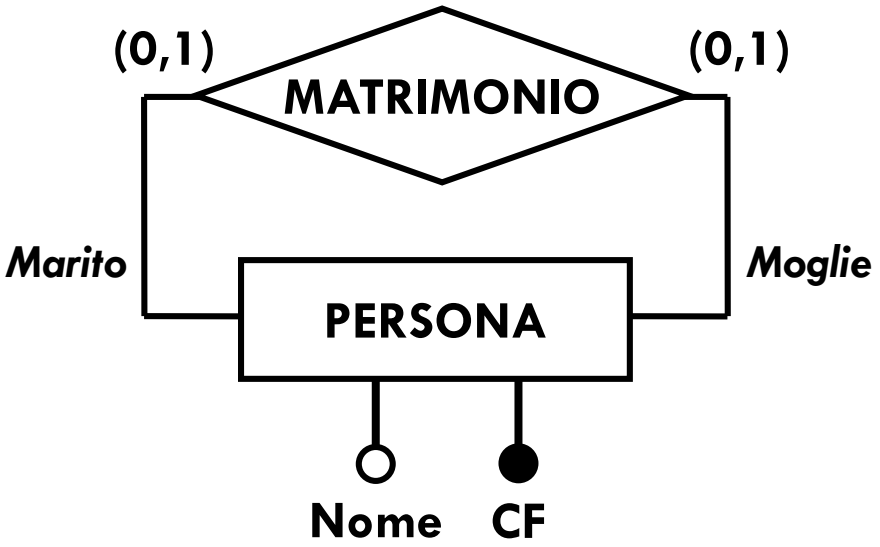
- La traduzione con una sola relazione corrisponde a un accorpamento di entità:
  - Se min-card(E1,R) = min-card(E2,R) = 1 si avranno due chiavi, entrambe senza valori nulli (la chiave primaria è "la più importante");
  - Se min-card(E1,R) = 0 e min-card(E2,R) =1 la chiave derivante da E2 ammetterà valori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



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

Associazioni ad anello uno a uno

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



Una relazione:

PERSONE(Codice, CFUomo*, NomeUomo*, CFDonna*, NomeDonna*)

Due relazioni:

PERSONE(CF, Nome)

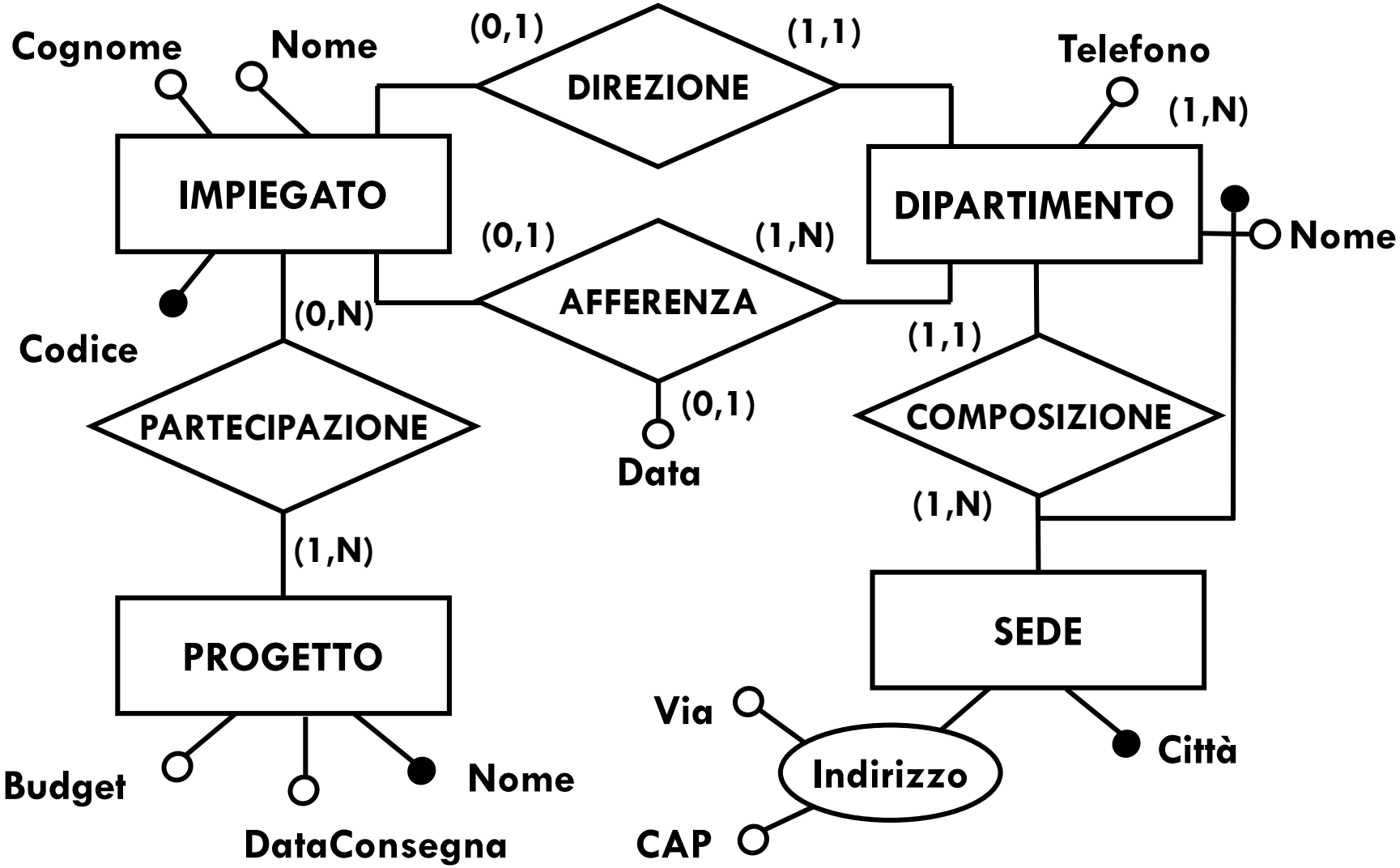
MATRIMONI(Marito, Moglie)

FK: Marito REFERENCES Persone

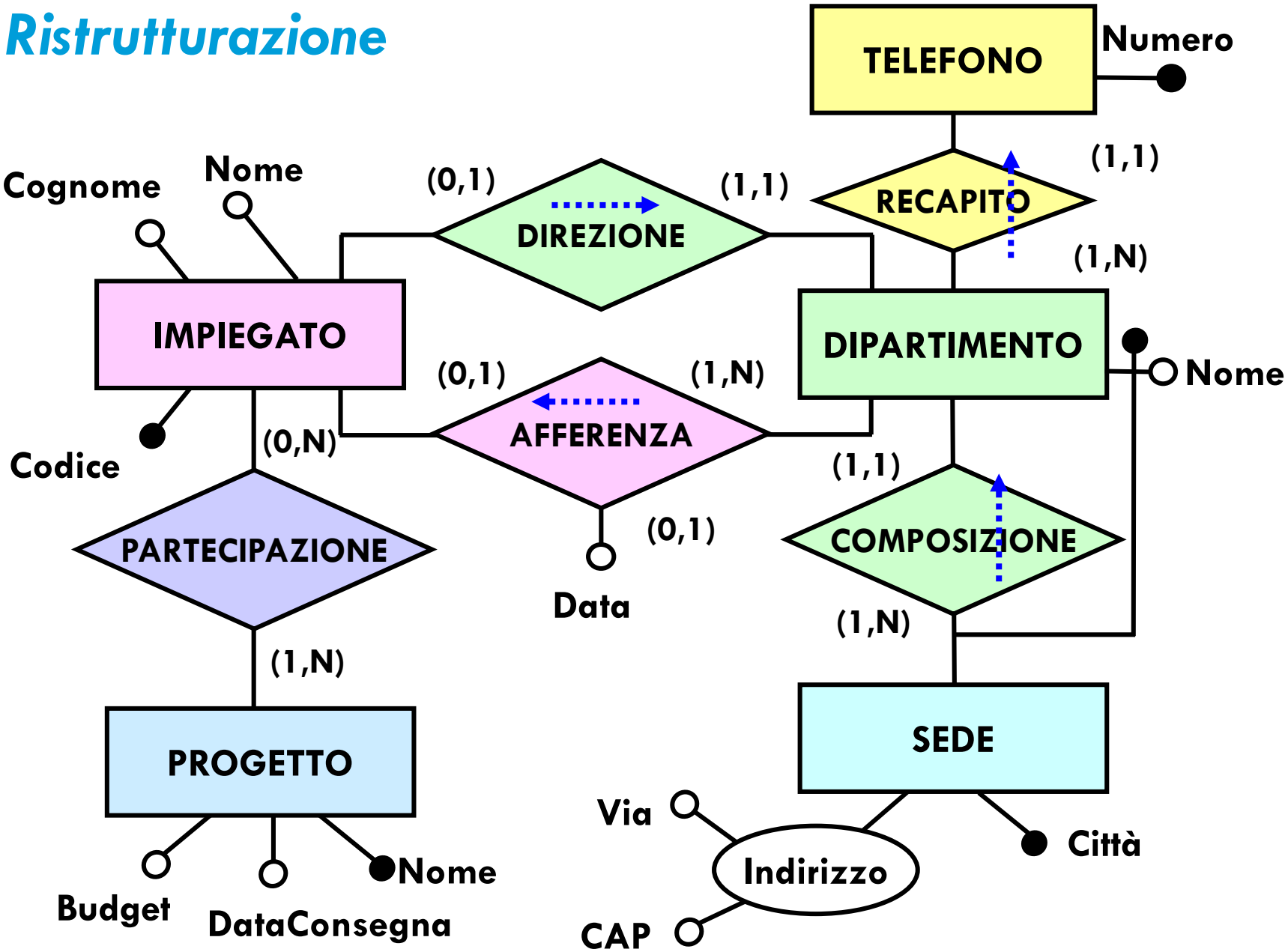
FK: Moglie REFERENCES Persone

Unique(Moglie)

Esempio di riferimento



Ristrutturazione



Schema logico relazionale

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

  SEDI(Città, Via, CAP)

  PROGETTI(Nome, Budget, DataConsegna)
- Anche l'associazione Partecipazione si traduce immediatamente:

  PARTECIPAZIONI(Impiegato, Progetto)

  FK: Impiegato REFERENCES Impiegati

  FK: Progetto REFERENCES Progetti
- L'entità Dipartimento si traduce importando l'identificatore di Sede e inglobando l'associazione Direzione:

  DIPARTIMENTI(Nome, Città, Direttore)

  FK: Città REFERENCES Sedi

  FK: Direttore REFERENCES Impiegati
- L'entità Telefono si traduce con una relazione che ingloba l'associazione Recapito

  TELEFONI(Numero, Nome, Città)

  FK: Nome, Città REFERENCES Dipartimenti
- Per tradurre l'associazione Afferenza, assumendo che siano pochi gli impiegati che non afferiscono a nessun dipartimento, si opta per una rappresentazione compatta

  IMPIEGATI(Codice, Nome, Cognome, NomeDip*, CittàDip*, Data*)

  FK: NomeDip, CittàDip REFERENCES Dipartimenti

Osservazioni finali

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

Domande?