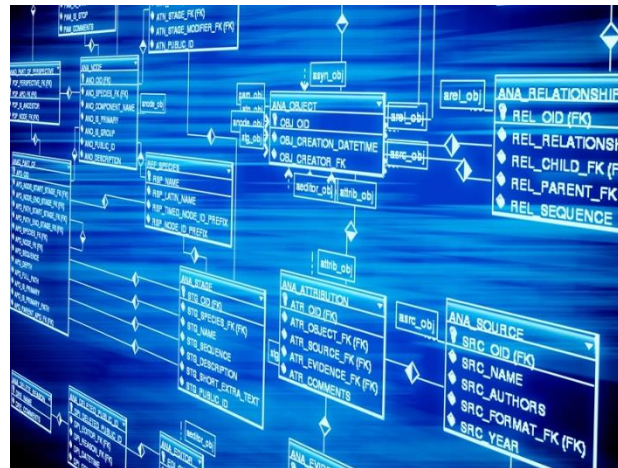


Politecnico di Milano - AA 2018-2019

Prof.ssa Sara Comai

Progettazione delle basi di dati: Progetto logico

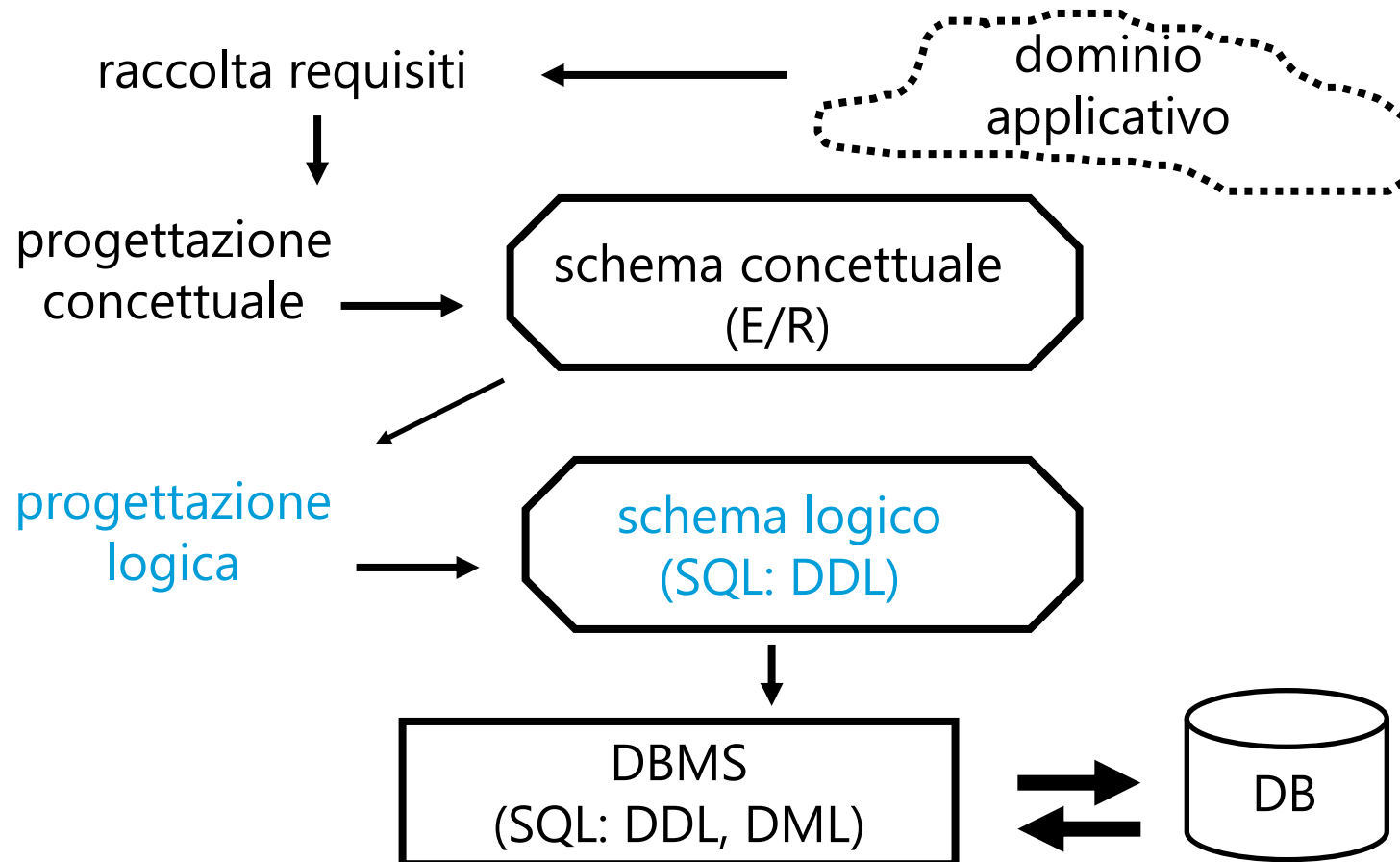


Progetto logico

Non esistono **DBMS** in grado di operare direttamente sui concetti di schemi E/R:

- è quindi necessario tradurli in **altri schemi** di dati (**logico relazionale** in queste lezioni)
- questa traduzione può essere eseguita in modo **semi-automatico**
- le **scelte alternative** devono tenere conto dell'efficienza dello **schema logico** risultante e delle operazioni da effettuare (derivanti da flussi e processi)

Processo di design



Fasi del progetto

1. Eliminazione delle gerarchie isa
2. Selezione delle chiavi primarie, eliminazione delle identificazioni esterne
3. Normalizzazione degli attributi composti o multipli
4. Traduzione di entità e associazioni in schemi di relazioni
5. Verifica di **normalizzazione**

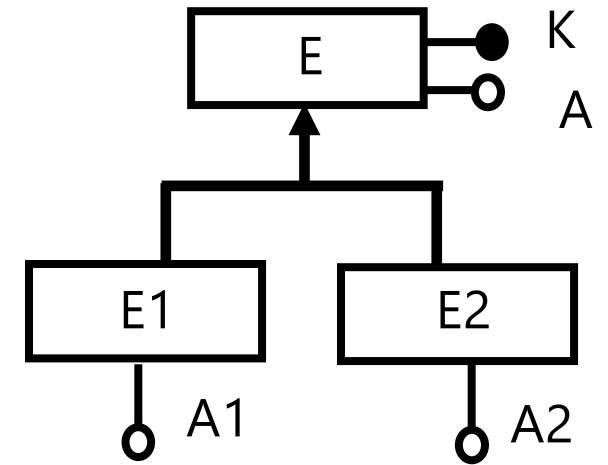
Eliminazione delle gerarchie

- Il modello relazionale non rappresenta le gerarchie: le gerarchie sono sostituite da entità e associazioni

Le possibili soluzioni sono:

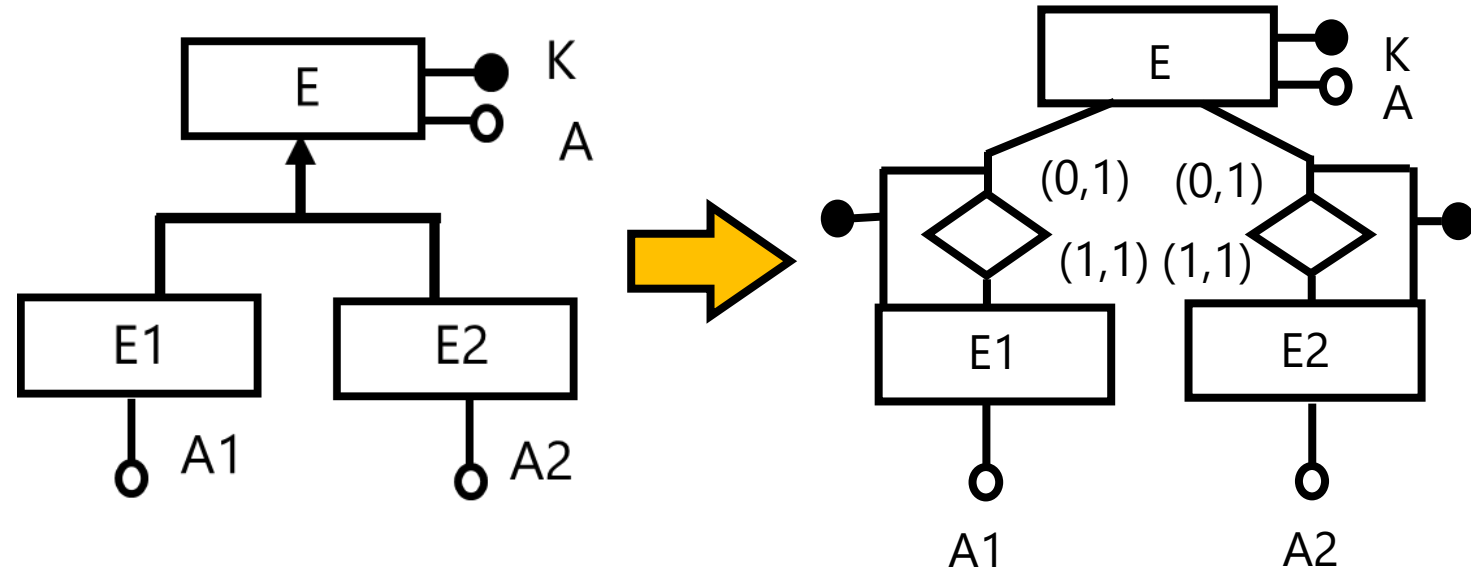
- 1. mantenimento delle entità con associazioni**
- 2. collasso verso l'alto**
- 3. collasso verso il basso**

- L'applicabilità e la convenienza delle soluzioni dipendono dalle proprietà di copertura e dalle operazioni previste



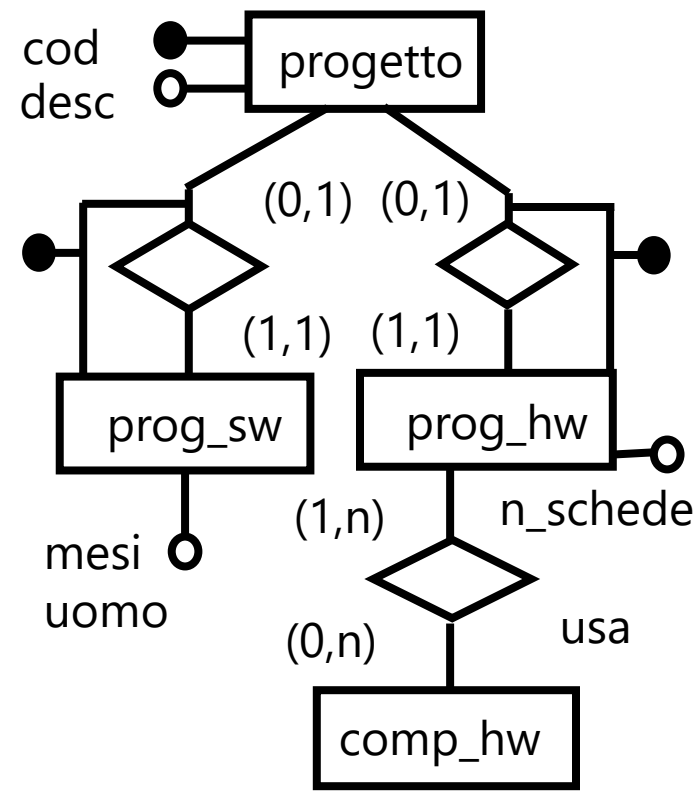
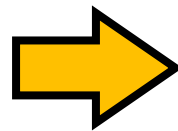
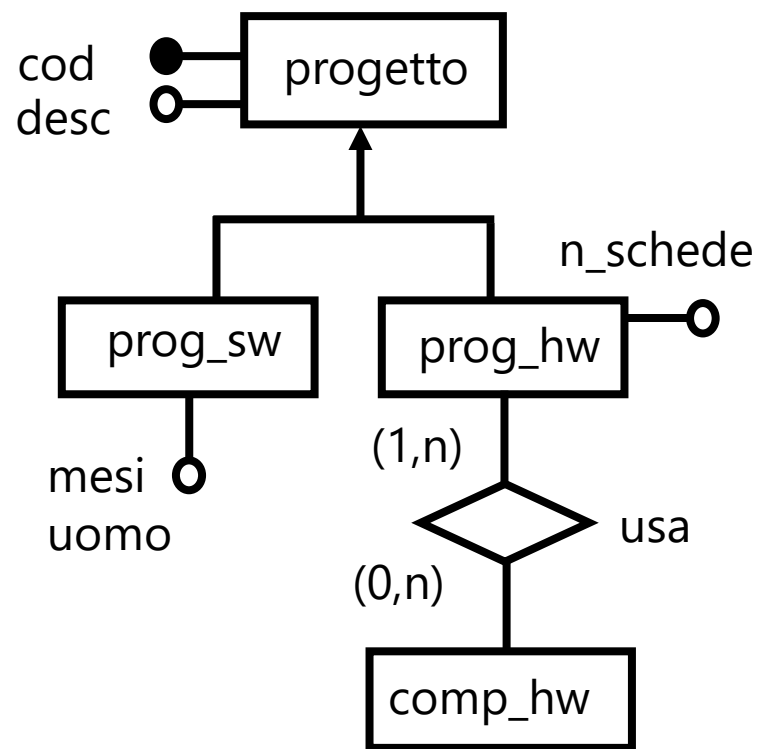
Mantenimento delle entità

- Tutte le entità vengono mantenute
- Le entità **figlie** sono in associazione con l'entità padre
- Le entità figlie sono **identificate esternamente** tramite l'associazione



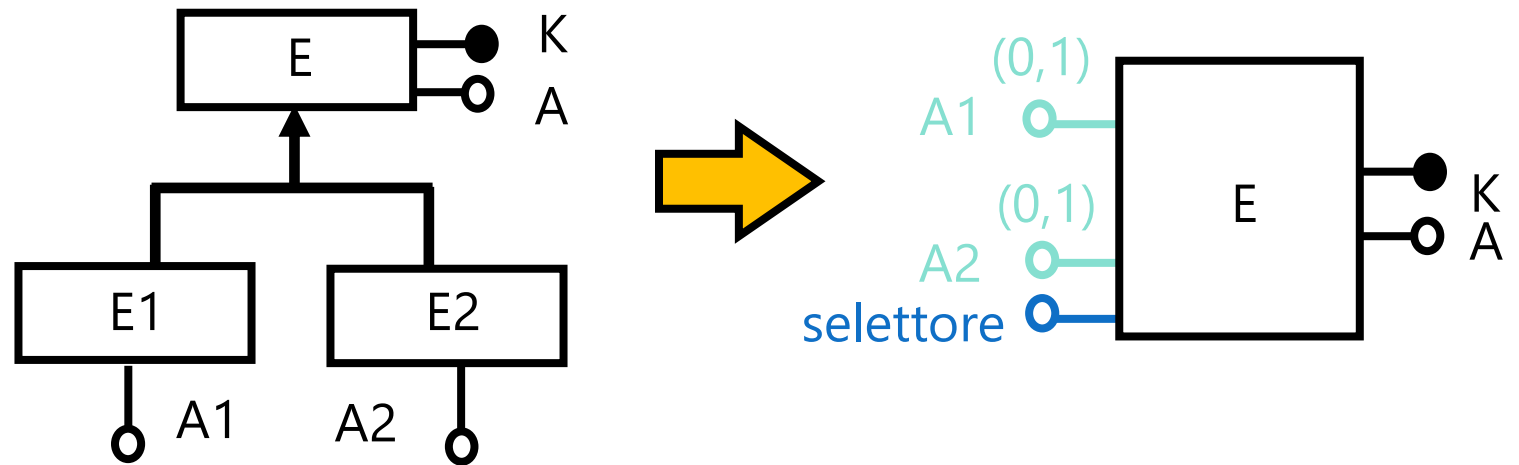
Questa soluzione è sempre possibile,
indipendentemente dalla copertura

Mantenimento entità - esempio



Eliminazione delle gerarchie

Il **collasso verso l'alto** riunisce tutte le entità figlie nell'entità padre.



selettore è un attributo che specifica se una istanza di E appartiene a una delle sottoentità.

Isa: collasso verso l'alto

Il **collasso verso l'alto** favorisce operazioni che consultano insieme gli attributi dell'entità padre e quelli di una entità figlia:

- in questo caso si accede a una sola entità, anziché a due attraverso una associazione.

Gli **attributi obbligatori** per le entità figlie **divengono opzionali** per il padre: si avrà una certa percentuale di **valori nulli**.

Isa: collasso verso l'alto

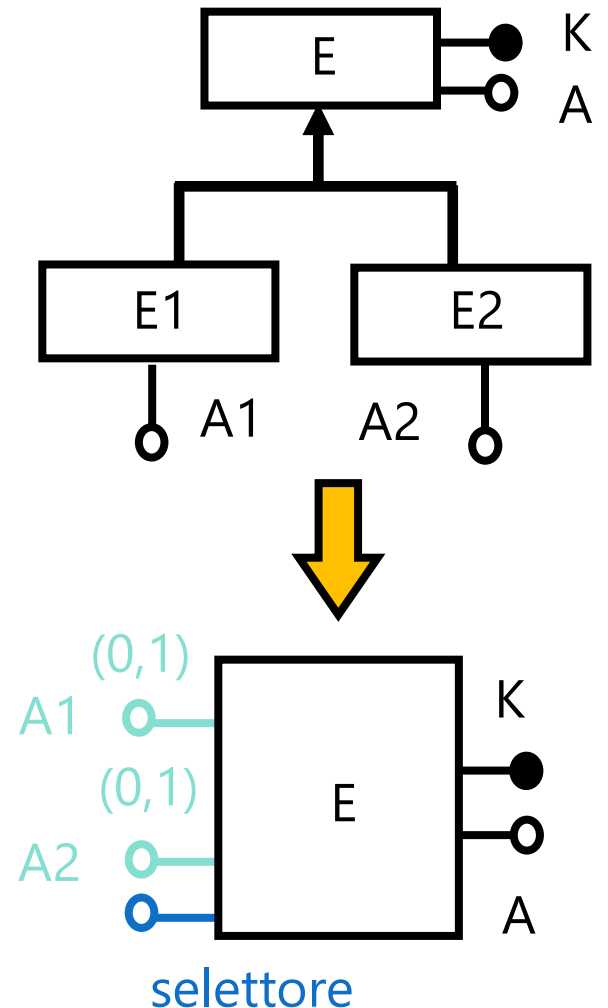
Copertura dell'ISA

totale esclusiva:

- selettore ha N valori,
- quante sono le sottoentità

parziale esclusiva:

- selettore ha $N+1$ valori;
- il valore in più serve per le istanze che non appartengono ad alcuna sottoentità

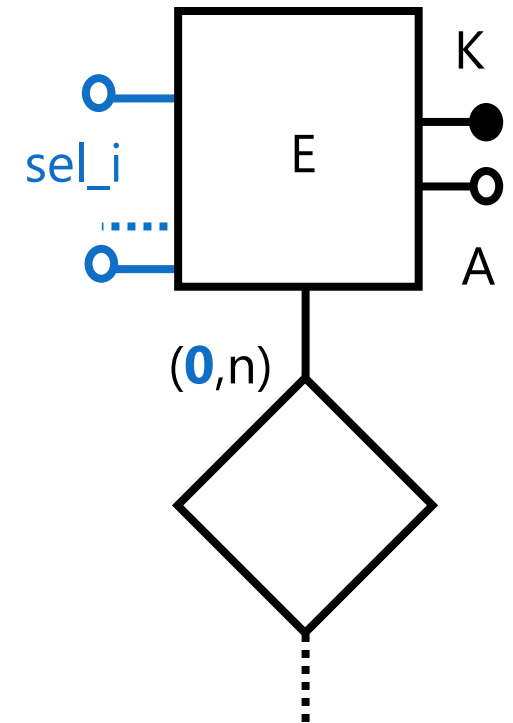


Isa: collasso verso l'alto

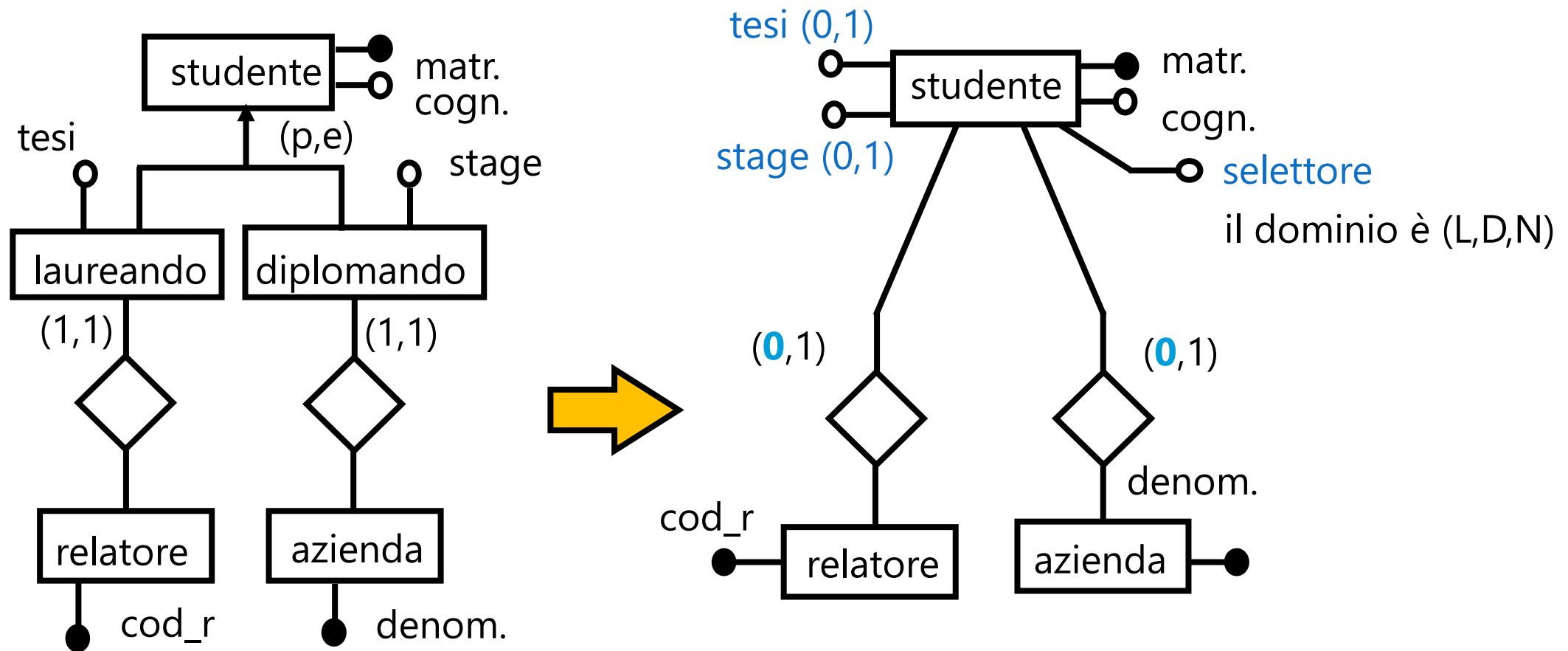
Copertura dell'ISA:

overlapping:

- Occorrono tanti **selettori booleani** quante sono le sottoentità
- **sel_i** è “vero” per ogni istanza di E che appartiene a E_i
- Se la copertura è **parziale** i selettori possono essere tutti “falsi”
- **Le associazioni** connesse alle sottoentità si trasportano su E, le eventuali cardinalità minime diventano 0



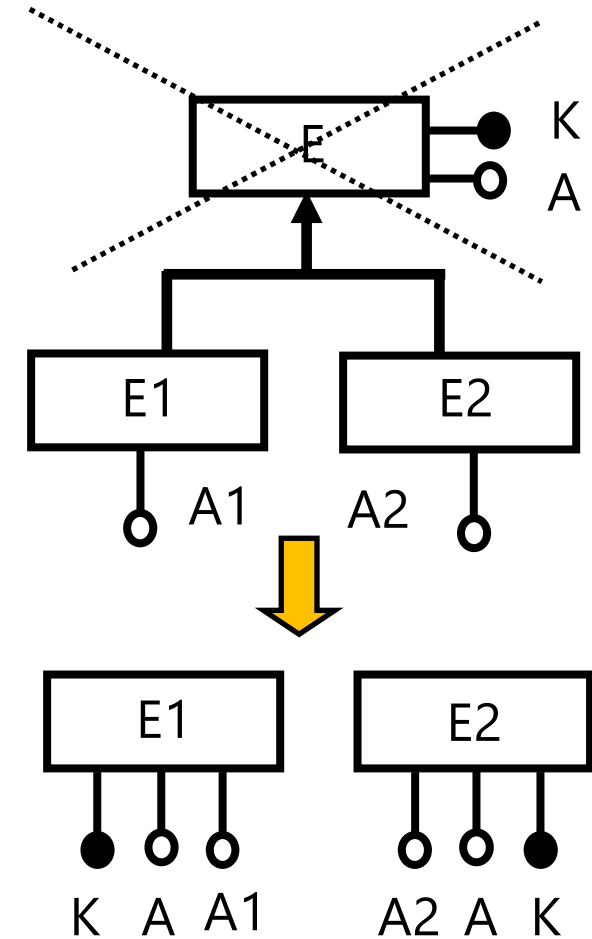
Isa: collasso verso l'alto



Isa: collasso verso il basso

Collasso verso il basso:

- si elimina l'entità **padre** trasferendone gli **attributi** su tutte le entità figlie:
- Un'associazione del padre è **replicata**, tante volte quante sono le entità figlie



Isa: collasso verso il basso

- La soluzione è interessante in presenza di molti attributi di specializzazione (con il collasso verso l'alto si avrebbe un eccesso di valori nulli)
- Favorisce le operazioni in cui si accede separatamente alle entità figlie

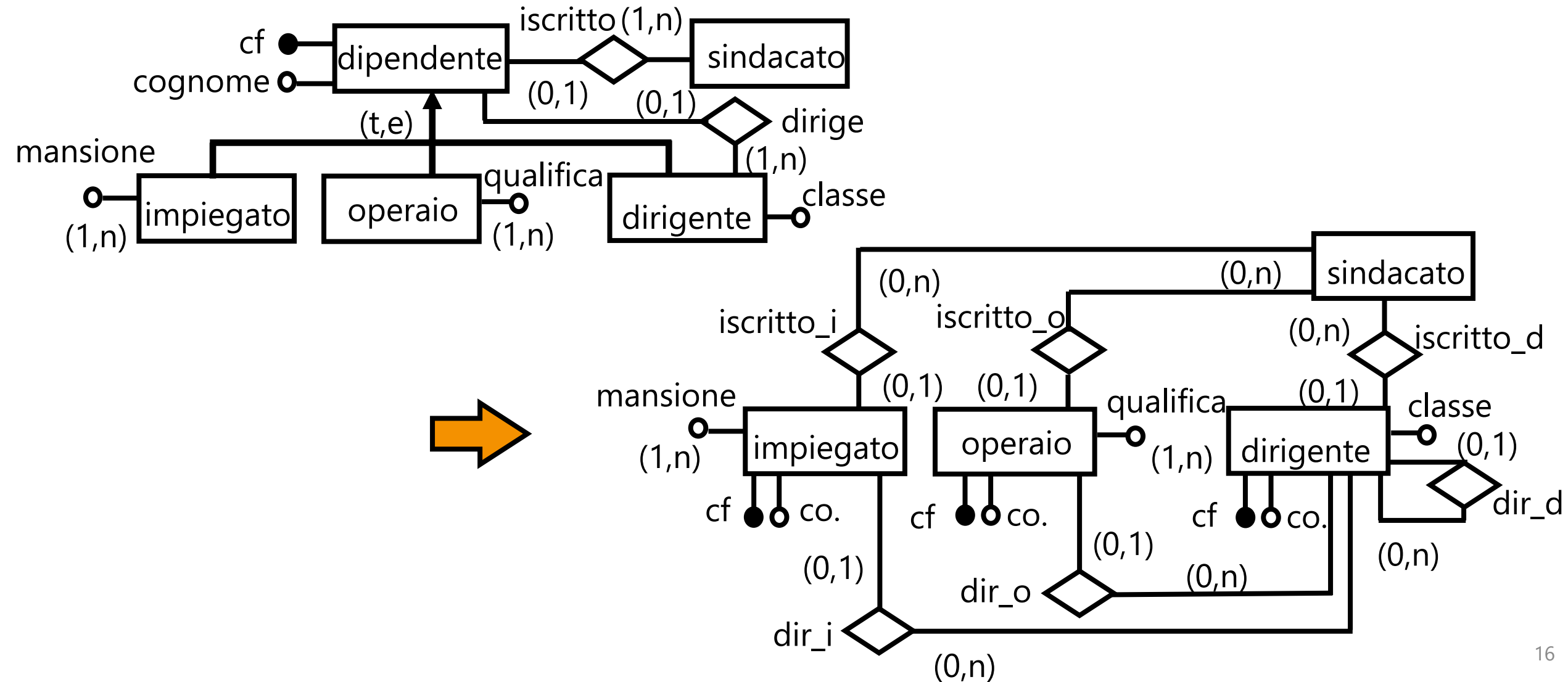
Limiti di applicabilità:

- se la copertura è **parziale**, **non si può fare**:

dove mettere gli E che non sono né E1, né E2 ?

- se la copertura è **overlapping**, introduce **ridondanza**: per una istanza presente sia in E1 che in E2 si rappresentano due volte gli attributi di E

Collasso verso il basso: esempio



Scelta della chiave primaria

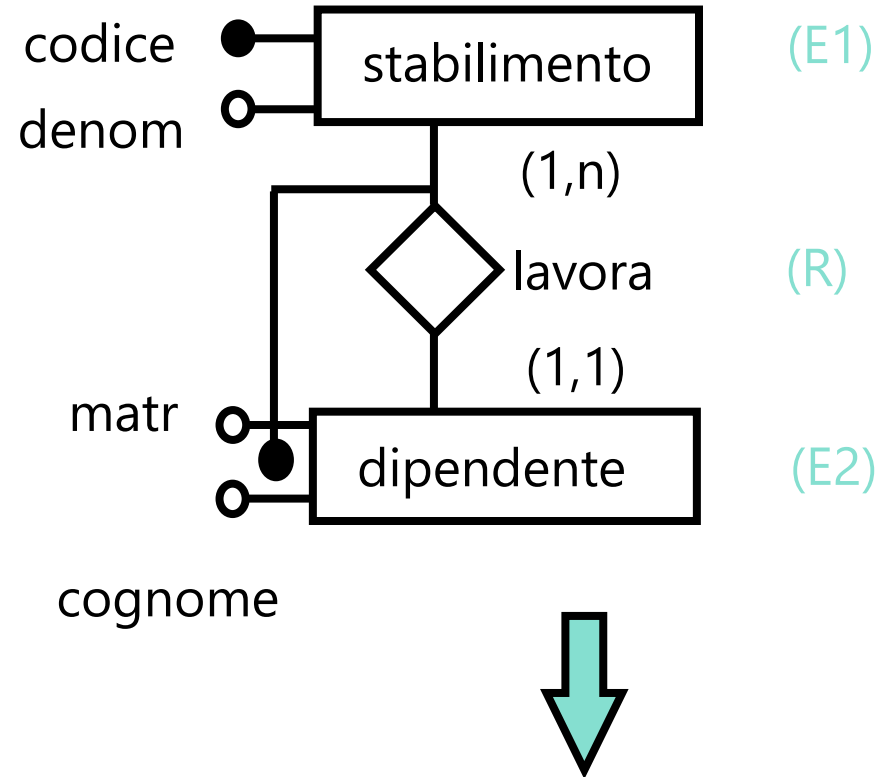
È necessario che tra i diversi **identificatori** di una entità venga designata una **chiave primaria**

Criteri euristici di scelta:

1. scegliere la chiave che è usata più frequentemente per accedere all'entità
2. si preferiscono chiavi semplici a chiavi composte, interne anziché esterne

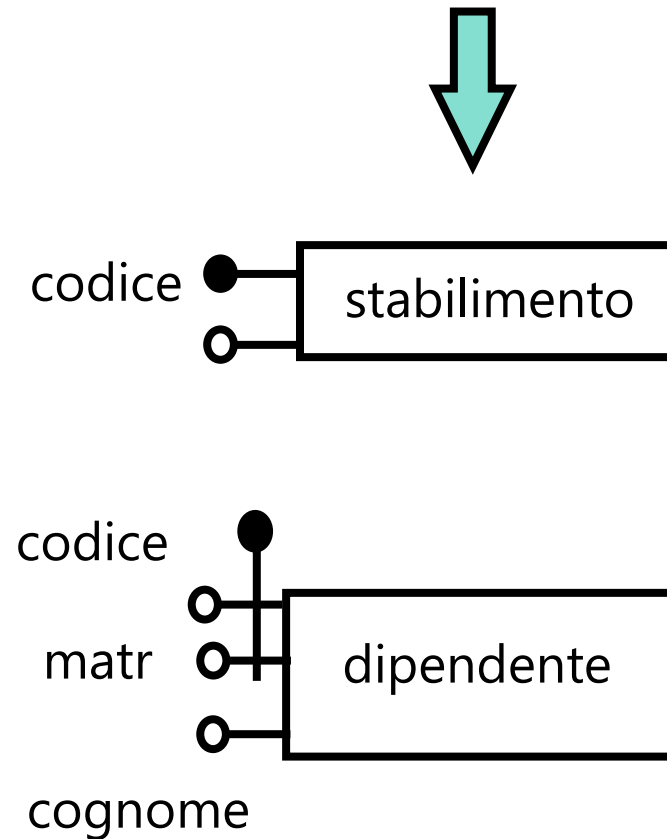
Identificatori esterni

Una componente di identificazione esterna di una entità E2 da una entità E1 attraverso una associazione R comporta il **trasporto della chiave primaria** di E1 su E2

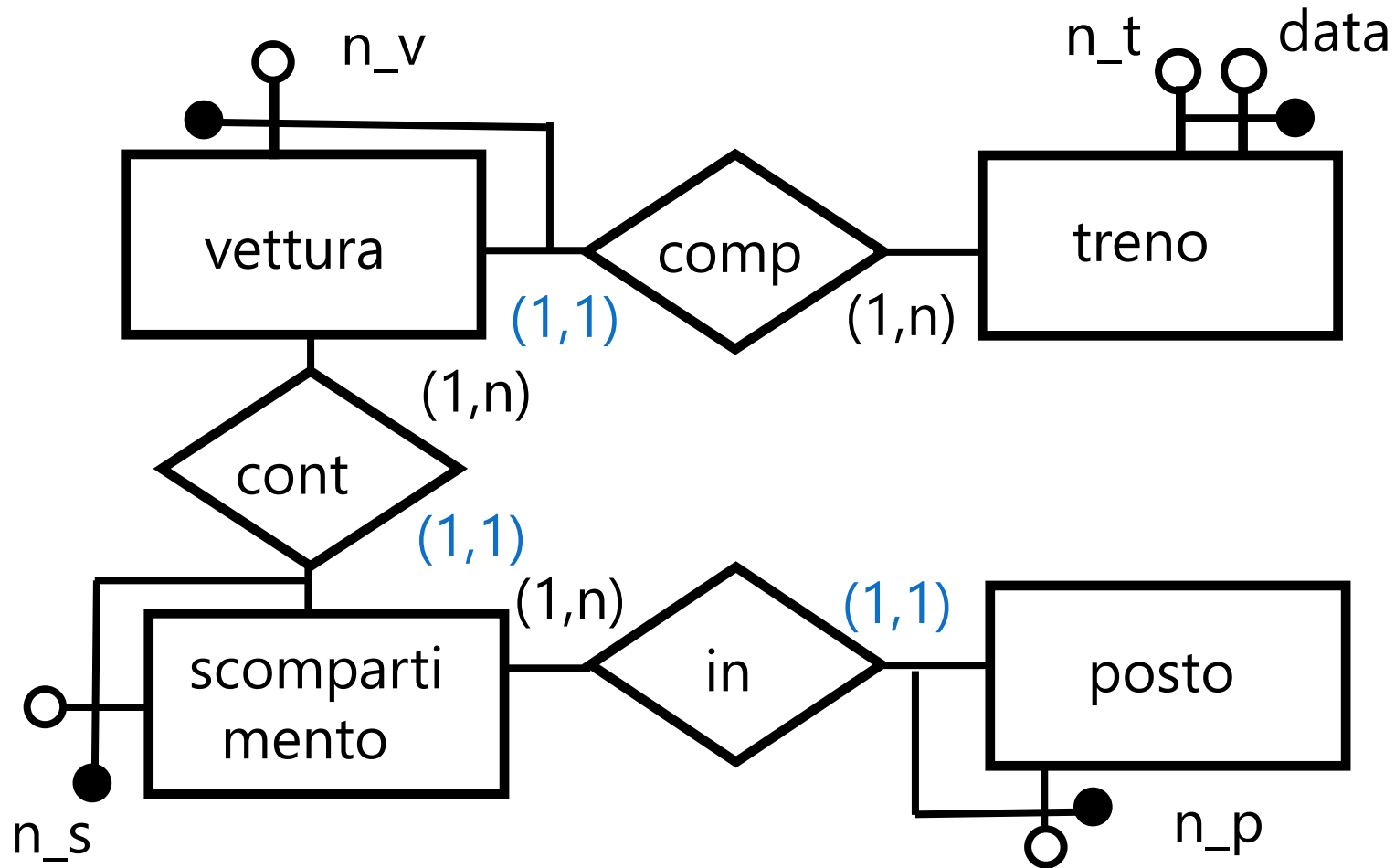


Identificatori esterni

- In questo modo l'associazione è rappresentata attraverso la chiave e può essere eliminata
- La chiave trasportata è chiave esterna
- In presenza di più identificazioni in cascata, è necessario iniziare la propagazione dall'entità che non ha identificazioni esterne



Esempio: posti treno



TRENO (N t, Data)

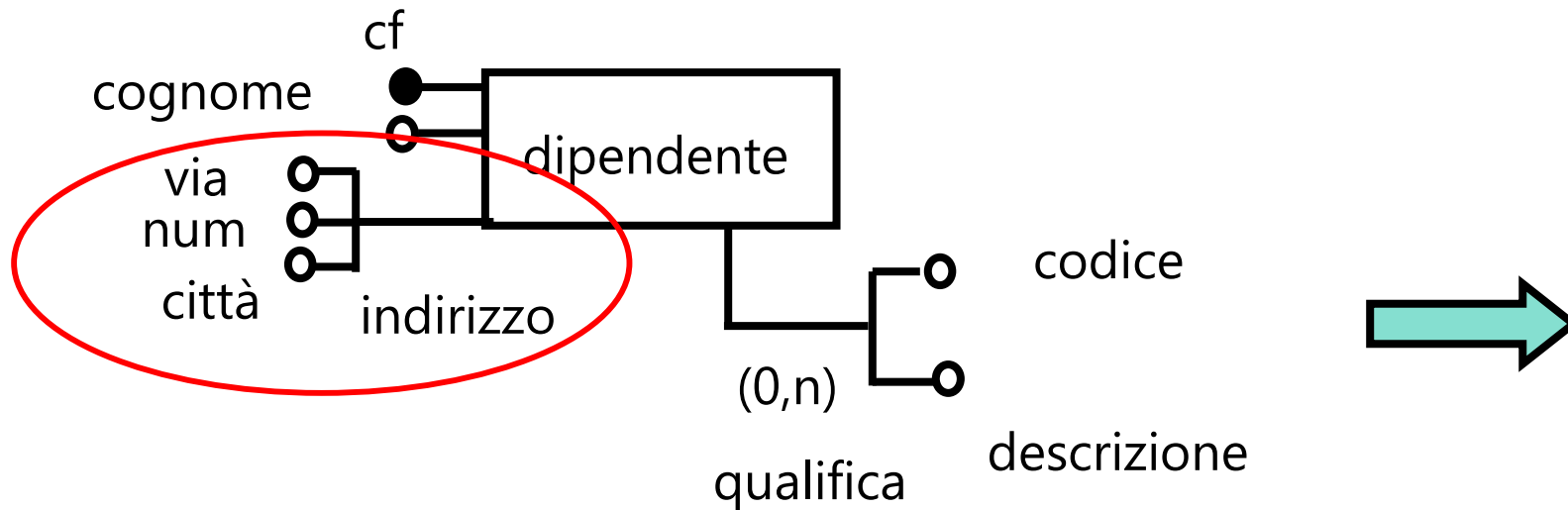
VETTURA (N t, Data, N v)

SCOMPARTIM (N t, Data, N v, N s)

POSTO (N t, Data, N v, N s, N p)

Attributi composti/ripetuti

Le relazioni non possono contenere attributi **composti** o attributi **ripetuti**, ma solamente attributi **"atomici"**

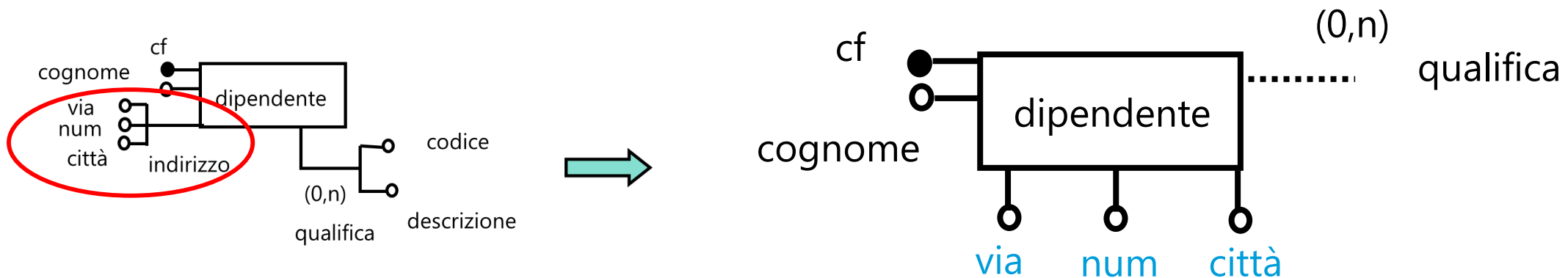


Attributi composti

Due possibili soluzioni:

1. **eliminare l'attributo composto** e considerare i suoi componenti come attributi semplici

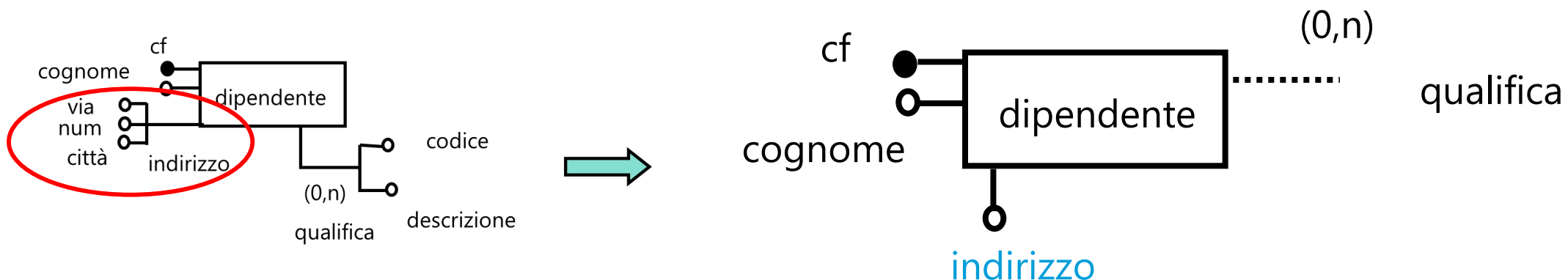
- CONTRO: si perde la visione unitaria
- PRO: si mantiene l'articolazione dei componenti.



Attributi composti

2. Eliminare i componenti e considerare l'attributo come semplice:

- in questo modo lo schema risulta semplificato, perdendo parte della struttura

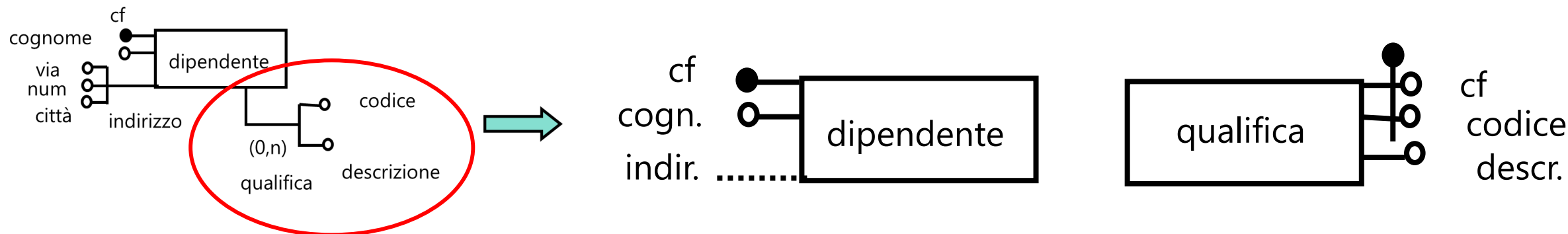


Attributi ripetuti

La definizione di relazione **impone** che, se una entità E ha un attributo A ripetuto, **si crei una nuova entità** che contenga l'attributo e sia collegata a E:

Caso a) - un valore può comparire una volta sola nella ripetizione:

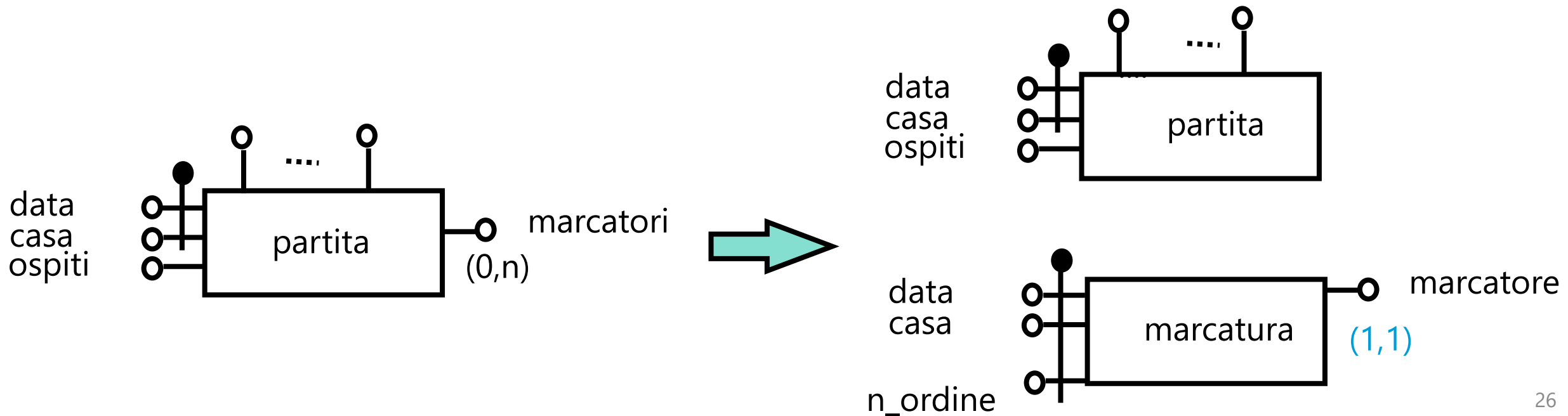
la nuova entità EA ha l'identificatore **composto** dall'identificatore di E più l'attributo A



Attributi ripetuti

Caso b) - un valore può comparire più volte nella ripetizione:

la nuova entità EA ha l'identificatore composto dall'identificatore di E più un valore identificante sintetico (ad esempio, un numero d'ordine)



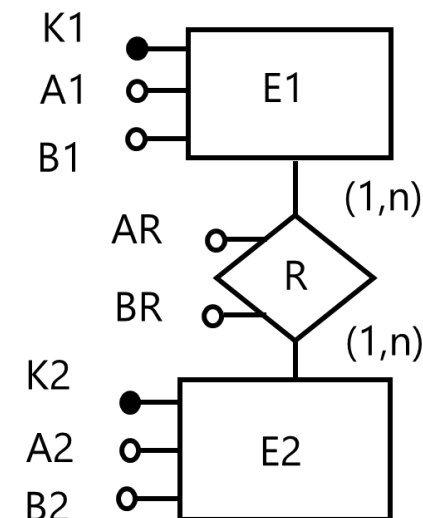
Traduzione standard di entità e relazioni

Ogni **entità** è tradotta con **una relazione** con gli stessi attributi:

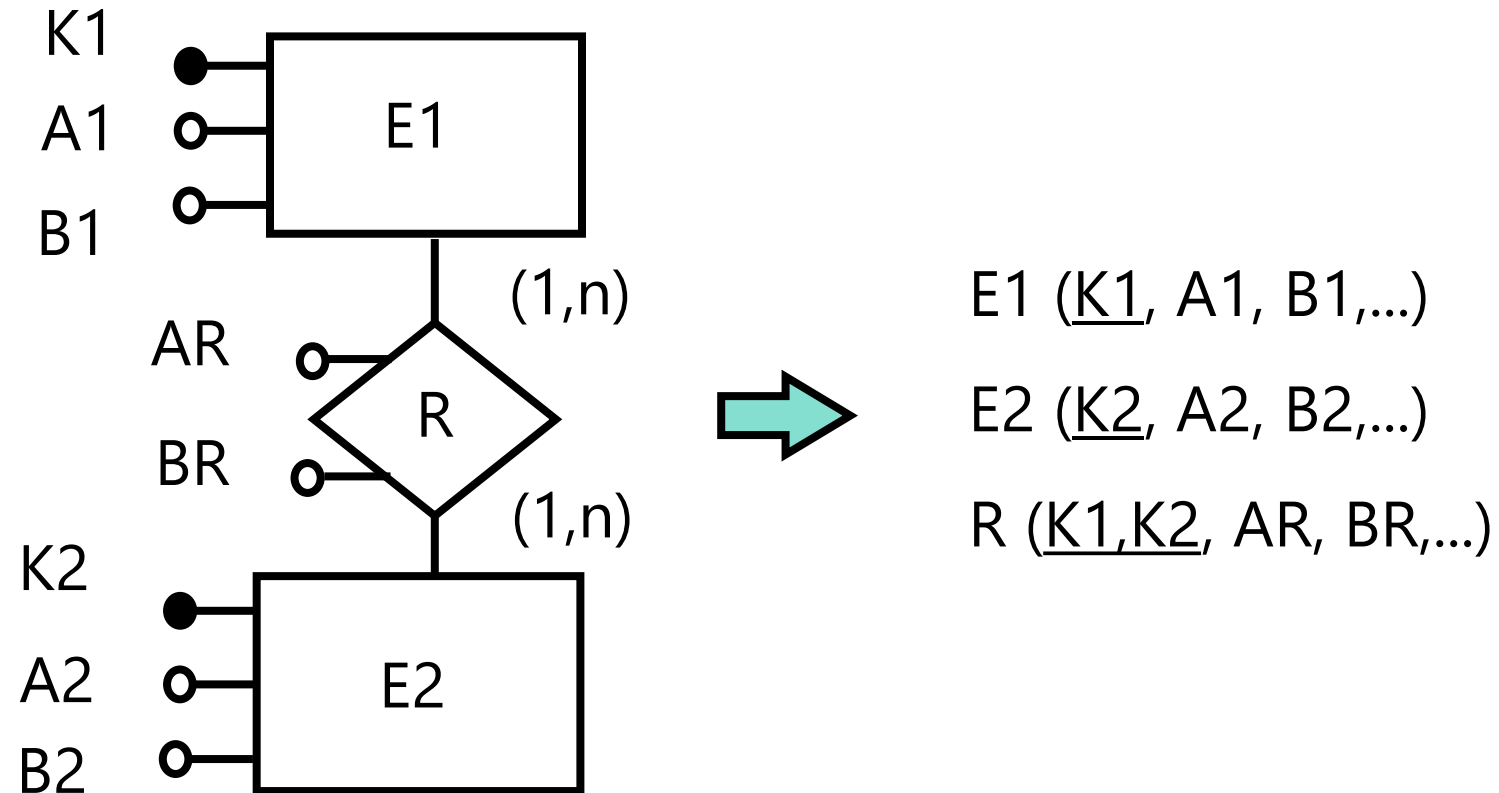
- la **chiave** è l'identificatore dell'entità stessa

Ogni **associazione** è tradotta con **una relazione** con gli stessi attributi, cui si aggiungono gli identificatori di tutte le entità che essa collega:

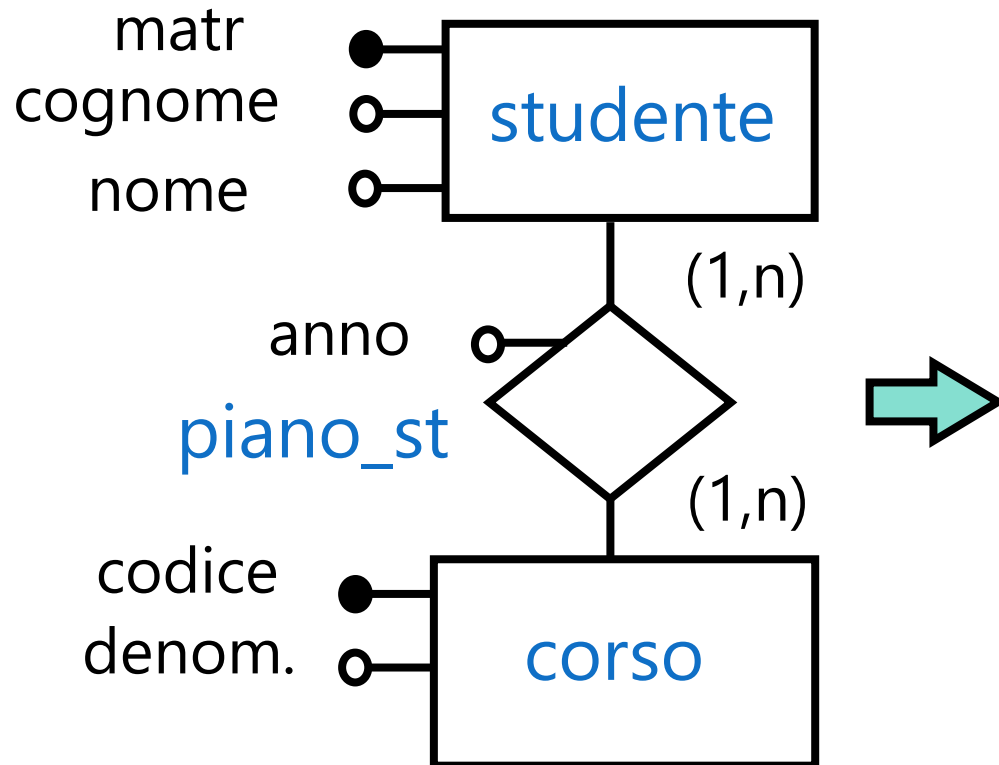
- la **chiave** è composta dalle chiavi delle entità collegate



Traduzione standard



Traduzione standard: esempio



in SQL...

```
CREATE TABLE STUDENTE (MATR ... PRIMARY KEY, NOME  
..., ... );
```

```
CREATE TABLE CORSO (CODICE... PRIMARY KEY, DENOM ...  
, ...);
```

```
CREATE TABLE PIANO_ST (MATR..., CODICE..., ANNO...  
  
PRIMARY KEY (MATR, CODICE),  
  
FOREIGN KEY (MATR) REFERENCES STUDENTE  
  
FOREIGN KEY (CODICE) REFERENCES CORSO);
```

Altre traduzioni

La **traduzione standard** è **sempre possibile** ed è l'**unica** possibilità per le associazioni N a M.

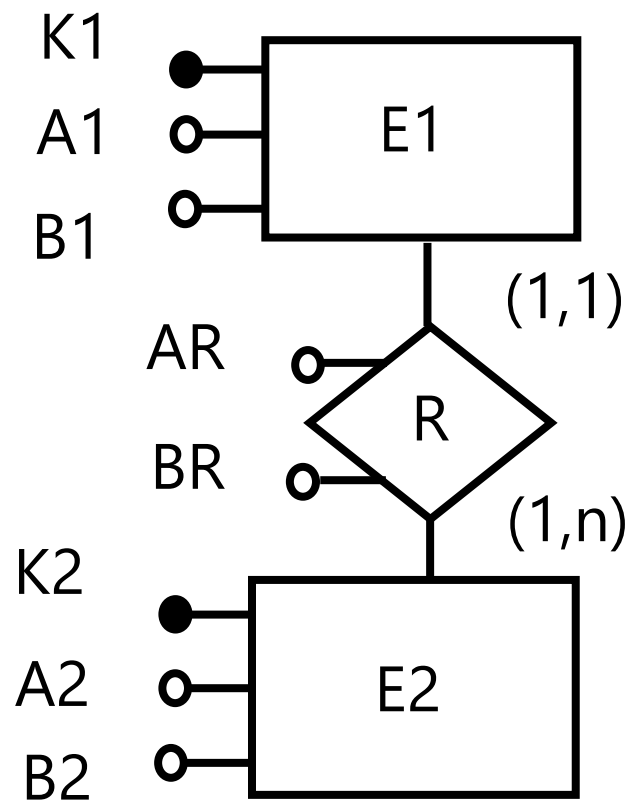
Altre forme di traduzione delle associazioni sono possibili **per altri casi di cardinalità** (1 a 1, 1 a N).

Le altre forme di traduzione **fondono** in una stessa relazione entità e associazioni.

Altre traduzioni

- Danno origine a un **minor numero** di relazioni e generano quindi uno schema più semplice
- Richiedono un minor numero di **join** per la **navigazione** attraverso un'associazione, ovvero per accedere alle istanze di entità connesse tramite l'associazione
- Penalizzano le operazioni che consultano **soltanto** gli attributi di una entità che è stata fusa

Associazione binaria 1 a N



Traduzione standard:

E1 (**K1**, A1, B1)

E2 (**K2**, A2, B2)

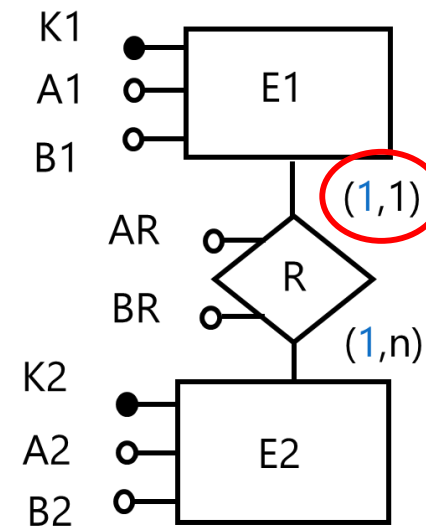
R (**K1, K2**, **AR, BR**)

Associazione binaria 1 a N

Se E1 partecipa con **cardinalità (1,1)** può essere **fusa** con l'associazione, ottenendo una soluzione a due relazioni:

$E1(\underline{K1}, A1, B1, K2, AR, BR)$

$E2(\underline{K2}, A2, B2)$



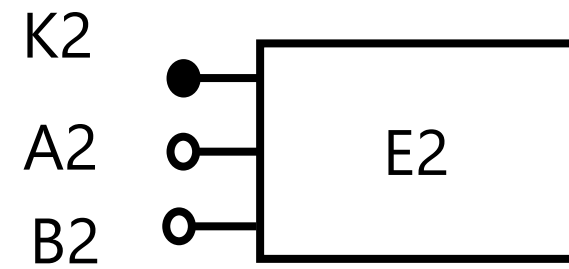
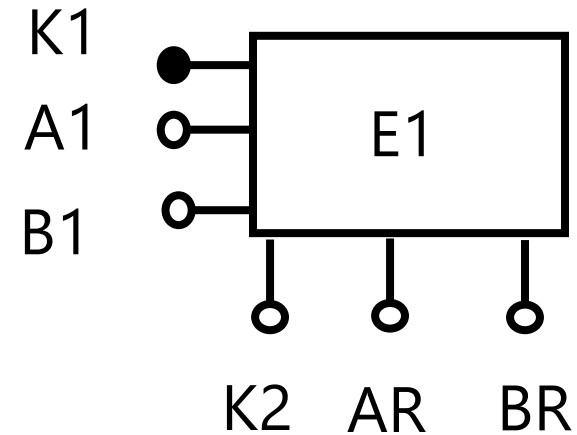
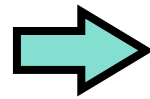
Se E1 partecipa con **cardinalità (0,1)** la soluzione a due relazioni **ha valori nulli in K2, AR, BR** per le istanze di E1 che non partecipano all'associazione

Associazione binaria 1 a N

$E1(\underline{K1}, A1, B1, K2, AR, BR)$

$E2(\underline{K2}, A2, B2)$

equivale a:

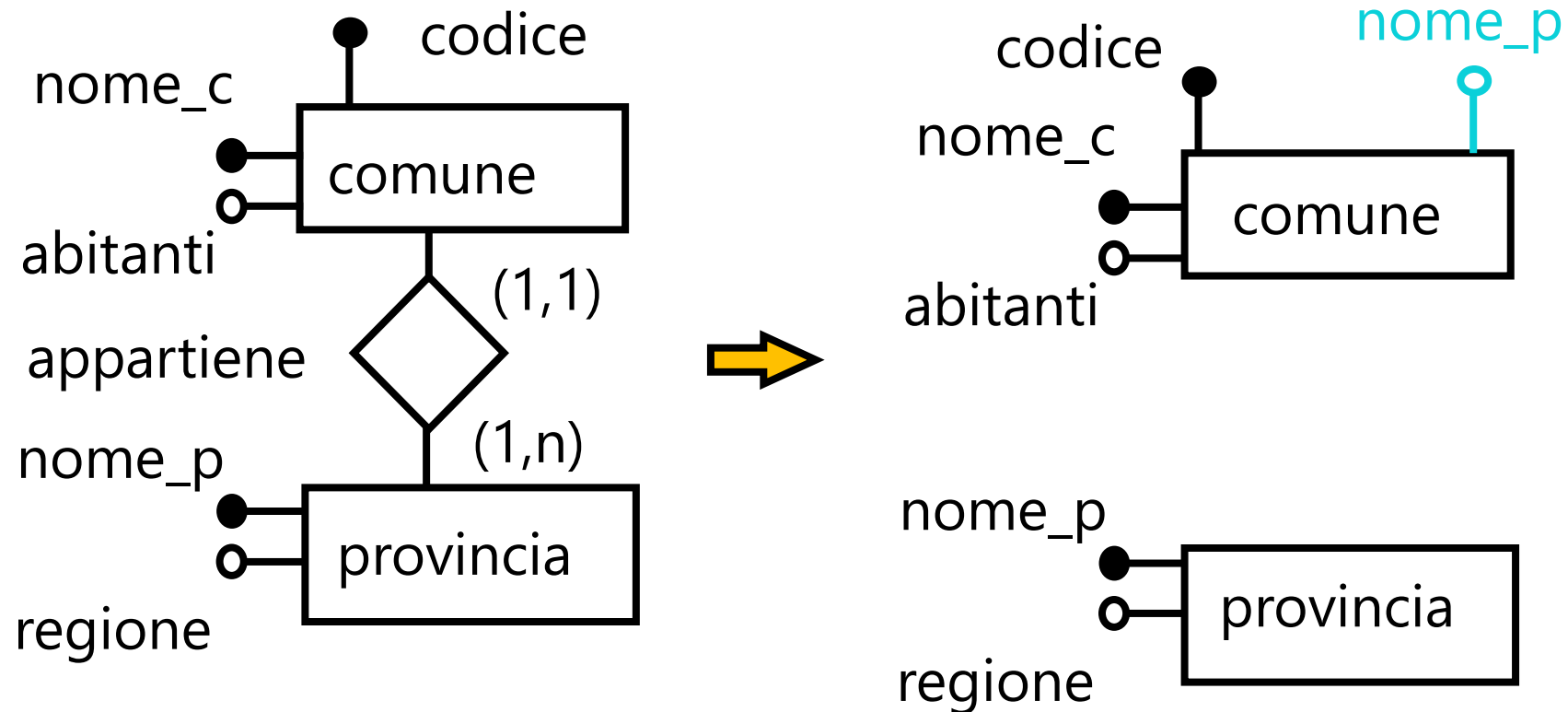


Associazione binaria 1 a N

- **Attenzione** : in questo caso, poiché la partecipazione di E1 è (0,1) oppure (1,1), si nota che ad un dato valore di K1 corrisponde uno e un sol valore di K2 (non è vero il contrario), quindi si può dire che **K1 implica K2** o, anche, che esiste una **dipendenza funzionale** da K1 a K2.
- Nella soluzione a 3 relazioni la chiave della relazione che traduce l'associazione è **riducibile a K1**:

E1(K1,A1,B1) E2(K2,A2,B2) R(K1,K2,AR,BR)

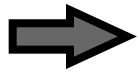
Ass. binaria 1 a N esempio



(senza attributi sull'associazione)

Ass. binaria 1 a N esempio

CREATE TABLE PROVINCIA



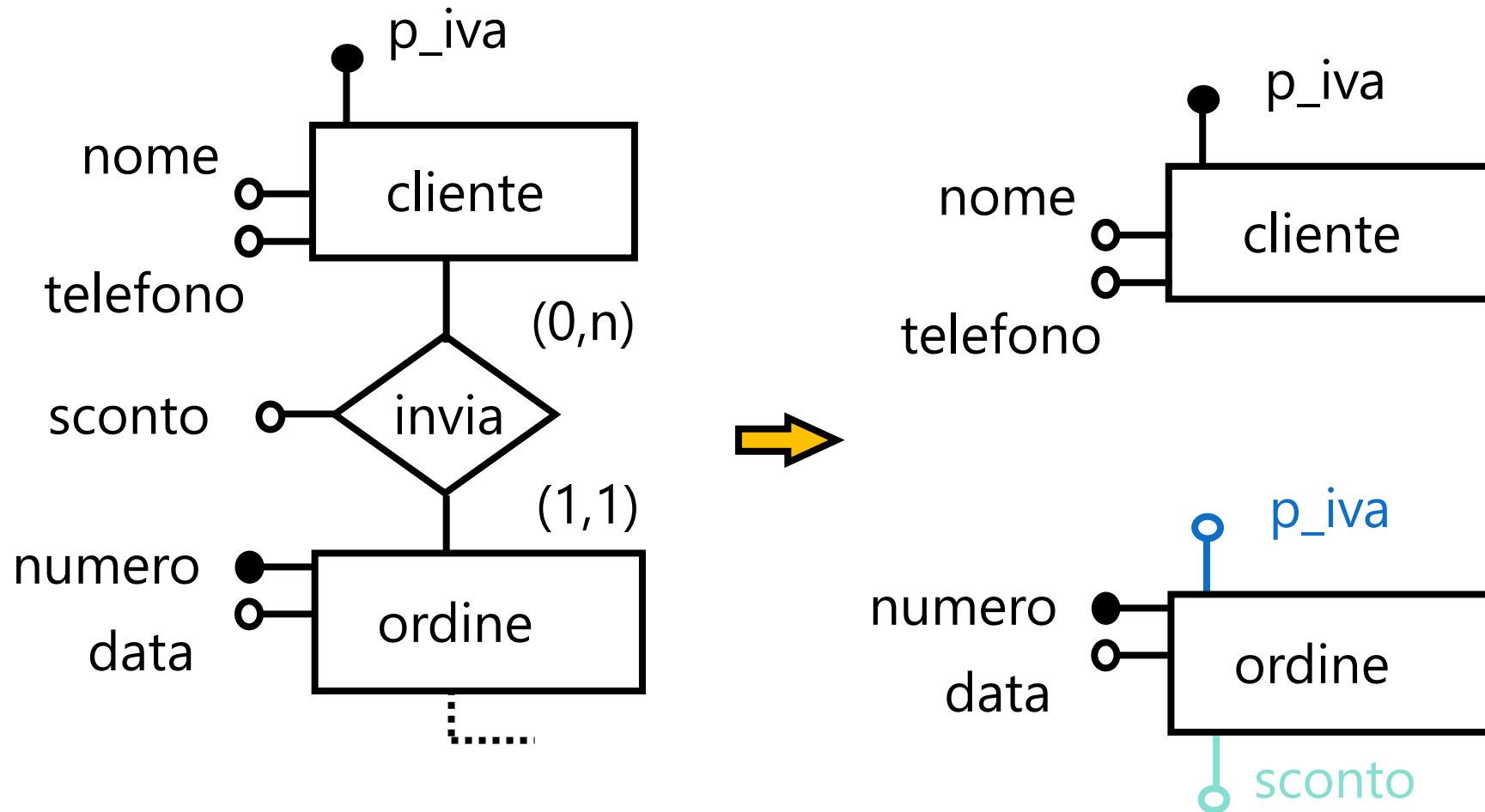
(NOME_P ... **PRIMARY KEY**,
REGIONE);

CREATE TABLE COMUNE

(CODICE ... **PRIMARY KEY**,
NOME_C ..., NOME_P

FOREIGN KEY NOME_P REFERENCES PROVINCIA);

Ass. binaria 1 a N esempio



(con attributi sull'associazione)

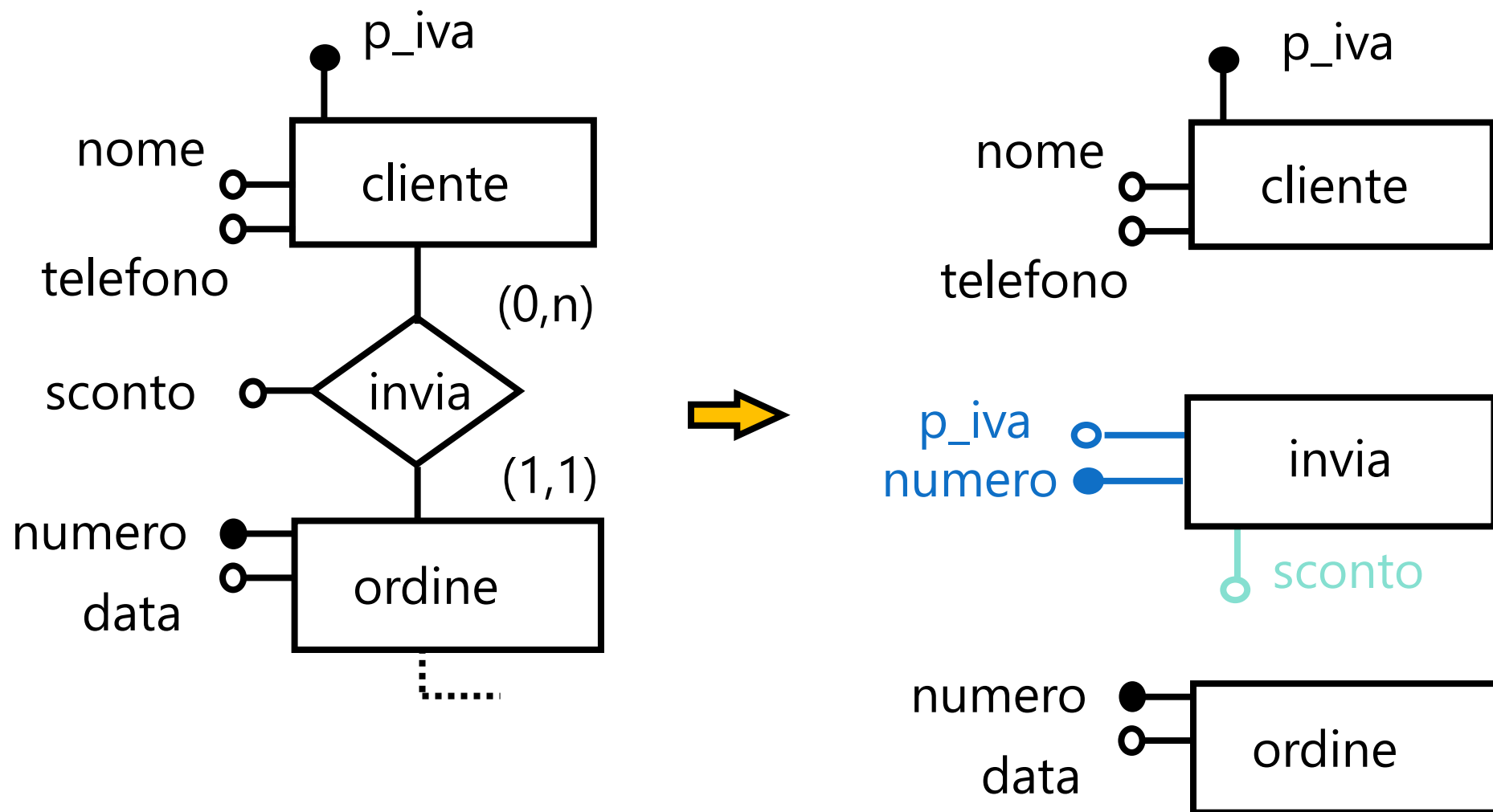
Ass. binaria 1 a N esempio

Traduzione con due relazioni:

```
CREATE TABLE CLIENTE (P_IVA..... PRIMARY KEY,  
NOME ...,TELEFONO ..., ....);
```

```
CREATE TABLE ORDINE (NUMERO ... PRIMARY KEY,  
DATA ... P_IVA ... NOT NULL, SCONTO ...,  
FOREIGN KEY P_IVA REFERENCES CLIENTE);
```

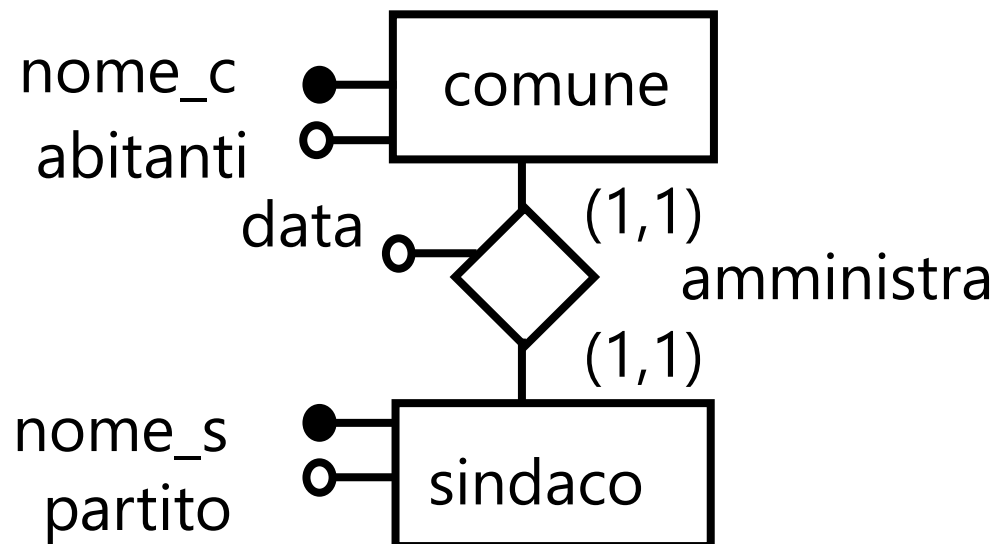
Ass. binaria 1 a N esempio (traduzione standard)



Associazione binaria 1 a 1

Traduzione con **una relazione**:

E12 (K1, A1, B1, K2, A2, B2, AR, BR)



CREATE TABLE AMMINISTRAZIONE

(NOME_C ... PRIMARY KEY,

ABITANTI ...,

NOME_S ... NOT NULL UNIQUE,

INDIRIZZO ..., DATA ...);

Se le cardinalità minime sono entrambe 1 la chiave può essere indifferentemente K1 o K2 si sceglierà quella più significativa.

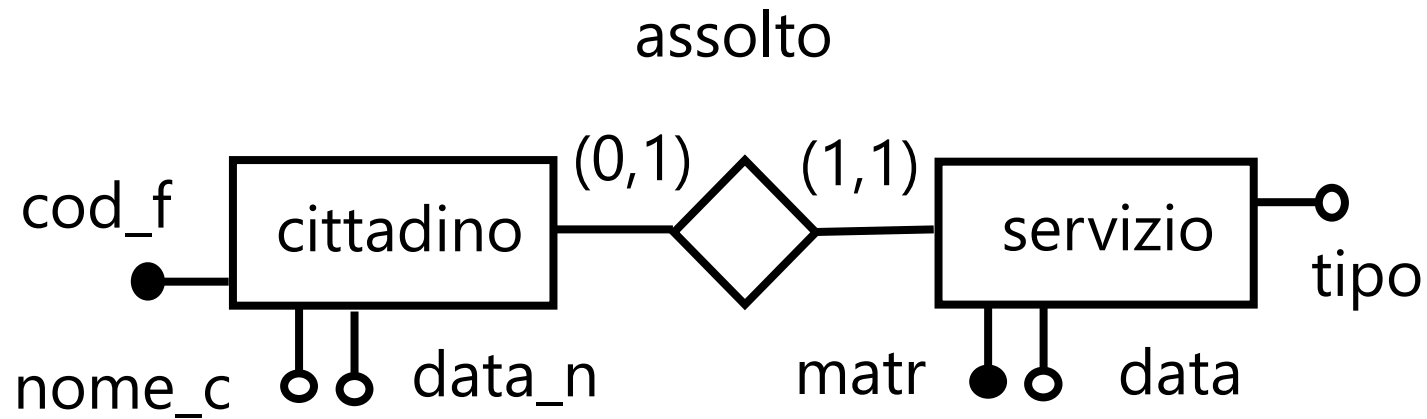
Associazione binaria 1 a 1

Se la cardinalità di E2 è **(0,1)** e quella di E1 è (1,1) allora la **chiave sarà K2**;

E2 è l'entità con maggior numero di istanze alcune della quali non si associano; ci saranno quindi valori nulli in corrispondenza di K1, K1 in questo caso non potrebbe essere scelta.

Se la cardinalità è **(0,1) da entrambe le parti** allora le **relazioni** saranno **due** per **l'impossibilità di assegnare la chiave** all'unica relazione a causa della presenza di valori nulli sia su K1 che su K2.

Associazione binaria 1 a 1



```
CREATE TABLE CITTADINO
```

```
(COD_F ... PRIMARY KEY, NOME_C ... NOT NULL,
```

```
INDIRIZZO ..., DATA_N ....., MATR ....., DATA..., TIPO .....);
```

Associazione binaria 1 a 1

Traduzione con **due relazioni**:

- l'associazione può essere **compattata** con **l'entità che partecipa obbligatoriamente** (una delle due se la partecipazione è obbligatoria per entrambe) la discussione sulla chiave è analoga al caso di traduzione con una relazione

E1 (K1, A1, B1,...)

E2 (K2, A2, B2,... K1, AR, BR)

Associazione binaria 1 a 1

Traduzione con **tre relazioni**

- la chiave della relazione che traduce l'associazione può essere indifferentemente K1 o K2, non ci sono problemi di valori nulli

E1 (K1, A1, B1,...)

E2 (K2, A2, B2,...)

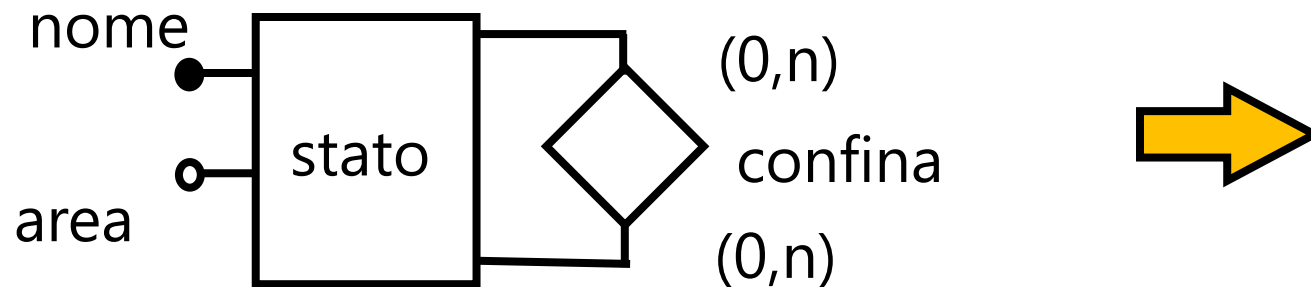
R (K1, K2, AR, BR,...)

Auto associazione N a M

Viene tradotta con:

una relazione per l'entità ed una per l'associazione

- quest'ultima contiene due volte la chiave dell'entità, è necessario però modificare i nomi degli attributi, per non avere omonimia.



Auto associazione N a M

CREATE TABLE STATO

(NOME ... PRIMARY KEY , AREA ...);

CREATE TABLE CONFINA

(STATO_A ... NOT NULL, STATO_B ... NOT NULL,

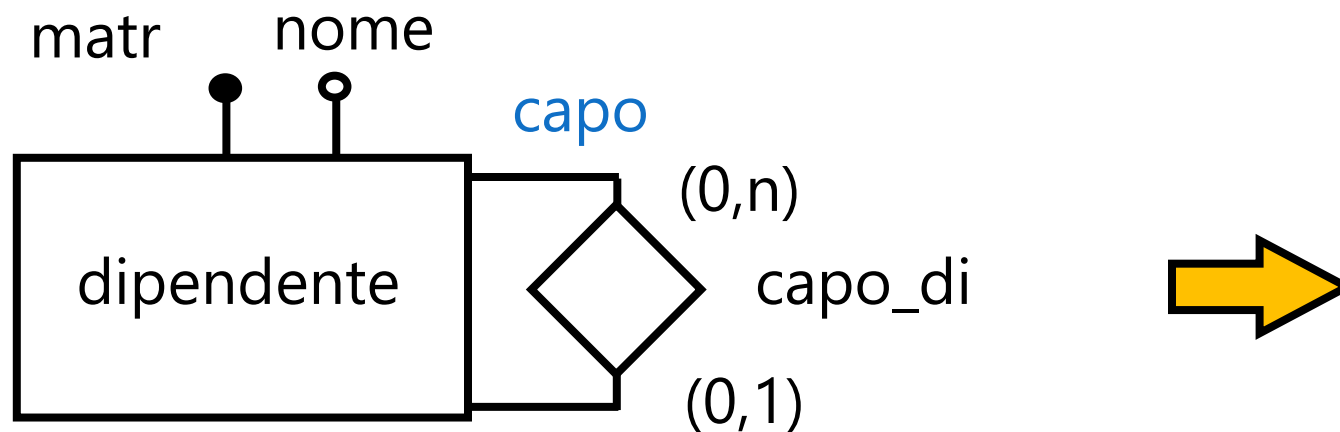
PRIMARY KEY (STATO_A, STATO_B)

FOREIGN KEY (STATO_A) REFERENCES STATO

FOREIGN KEY (STATO_B) REFERENCES STATO);

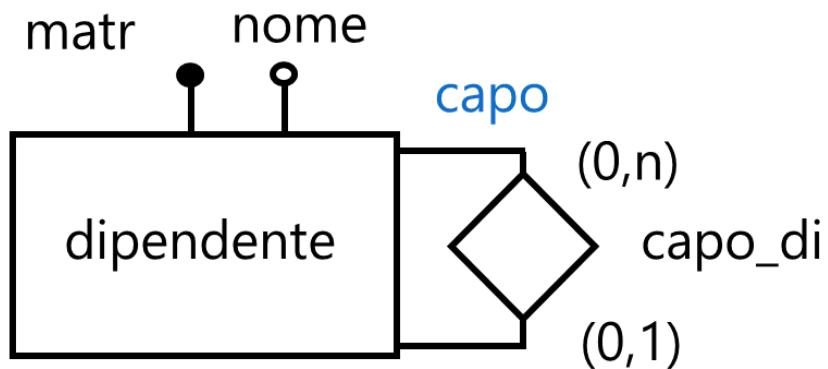
Auto associazione 1 a N

È traducibile con **una sola relazione** che contiene due volte l'attributo chiave: una volta come **chiave** ed una come **riferimento** all'istanza connessa, con nome diverso per specificare il **ruolo**.



Auto associazione 1 a N

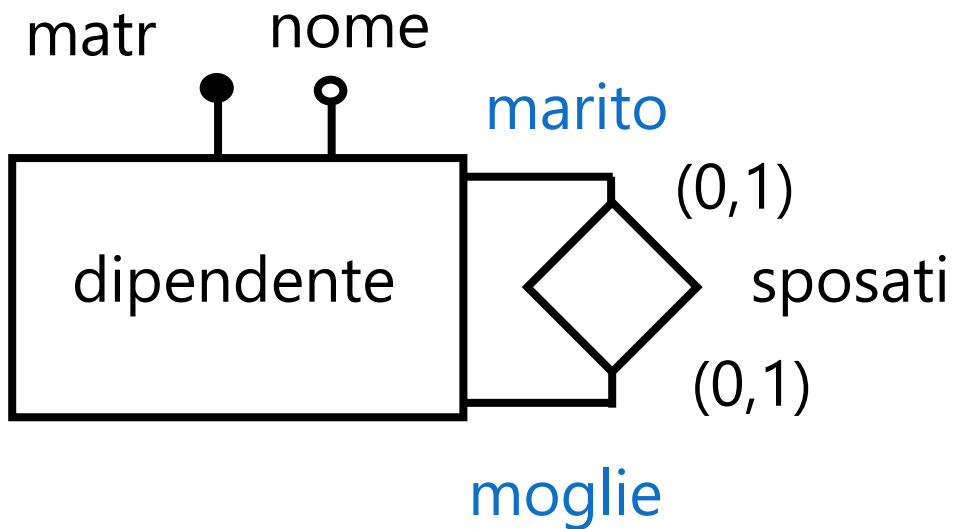
Nel caso di associazione 1 ad 1 il concetto di **ruolo** assume maggiore importanza:



```
CREATE TABLE DIPENDENTE
(MATR ... PRIMARY KEY, NOME ..., CAPO ...
FOREIGN KEY (CAPO) REFERENCES DIPENDENTE);
```

Auto associazione 1 a 1

Su **entrambi i rami** è bene specificare il **ruolo**: conviene la soluzione con due relazioni per evitare ridondanze, vincoli ed eccesso di valori nulli.



```
CREATE TABLE DIPENDENTE (MATR ... PRIMARY KEY,  
NOME ...);
```

```
CREATE TABLE SPOSATI
```

```
(MOGLIE ... PRIMARY KEY, MARITO ... NOT NULL
```

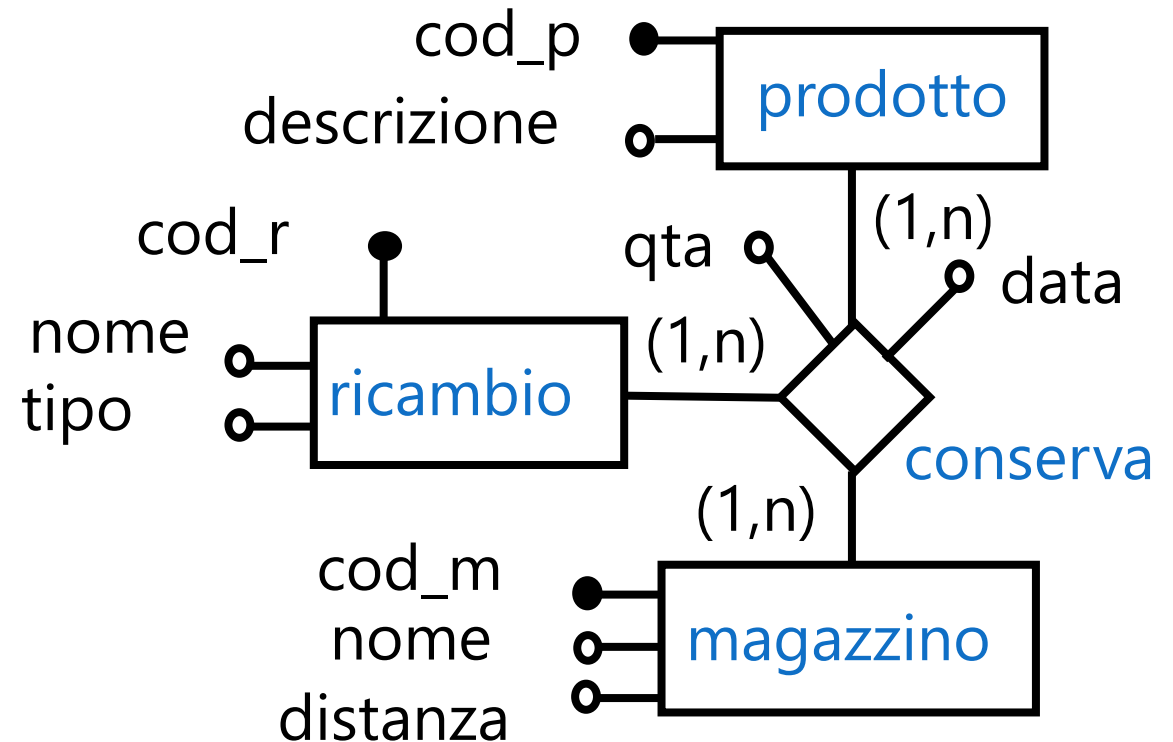
```
FOREIGN KEY (MOGLIE) REFERENCES DIPENDENTE
```

```
FOREIGN KEY (MARITO) REFERENCES DIPENDENTE);
```


Associazione n-aria

Segue la traduzione standard.

Talvolta, nella relazione che traduce l'associazione, la chiave ottenuta componendo le chiavi di tutte le entità partecipanti è una **superchiave**, cioè una chiave composta il cui set di componenti **non è minimale** (la chiave vera è un sottoinsieme).



Associazione n-aria

```
CREATE TABLE PRODOTTO (COD_P... PRIMARY KEY, DESCRIZIONE... );
```

```
CREATE TABLE RICAMBIO (COD_R ... PRIMARY KEY, NOME..., TIPO...);
```

```
CREATE TABLE MAGAZZINO (COD_M.... PRIMARY KEY, NOME ..., DISTANZA...);
```

L'associazione diventa:

```
CREATE TABLE CONSERVA (COD_P ... NOT NULL, COD_R... NOT NULL, COD_M... NOT NULL,  
DATA..., QTA ...
```

```
PRIMARY KEY (COD_P, COD_R, COD_M)
```

```
FOREIGN KEY (COD_P) REFERENCES PRODOTTO
```

```
FOREIGN KEY (COD_M) REFERENCES MAGAZZINO
```

```
FOREIGN KEY (COD_R) REFERENCES RICAMBIO);
```

Associazione n-aria

```
CREATE TABLE CONSERVA (COD_P ... NOT NULL,  
COD_R... NOT NULL, COD_M... NOT NULL,  
DATA... NOT NULL, QTA ...  
PRIMARY KEY (COD_P, COD_R, COD_M)  
FOREIGN KEY (COD_P) REFERENCES PRODOTTO  
FOREIGN KEY (COD_M) REFERENCES MAGAZZINO  
FOREIGN KEY (COD_R) REFERENCES RICAMBIO);
```



```
CREATE TABLE CONSERVA (COD_P ... NOT NULL,  
COD_R... NOT NULL, COD_M... NOT NULL,  
DATA... , QTA ...  
PRIMARY KEY (COD_P, COD_R)  
FOREIGN KEY (COD_P) REFERENCES PRODOTTO  
FOREIGN KEY (COD_R) REFERENCES RICAMBIO)  
FOREIGN KEY (COD_M) REFERENCES MAGAZZINO);
```

ipotizziamo che COD_M sia ridondante: un ricambio esiste in un solo magazzino, quindi COD_R è associato ad un solo COD_M, cioè determina COD_M.

COD_M non è più parte della chiave

Commento

- Nel caso precedente la **dipendenza** tra **magazzino e ricambio** non era stata espressa sulla associazione n-aria; abbiamo ipotizzato di scoprirla nella fase di progetto logico
- Se il progetto concettuale è ben fatto casi del genere non sono frequenti
- Il **ricontrollo delle chiavi** delle relazioni è quindi importante e se ne occupa la teoria della normalizzazione