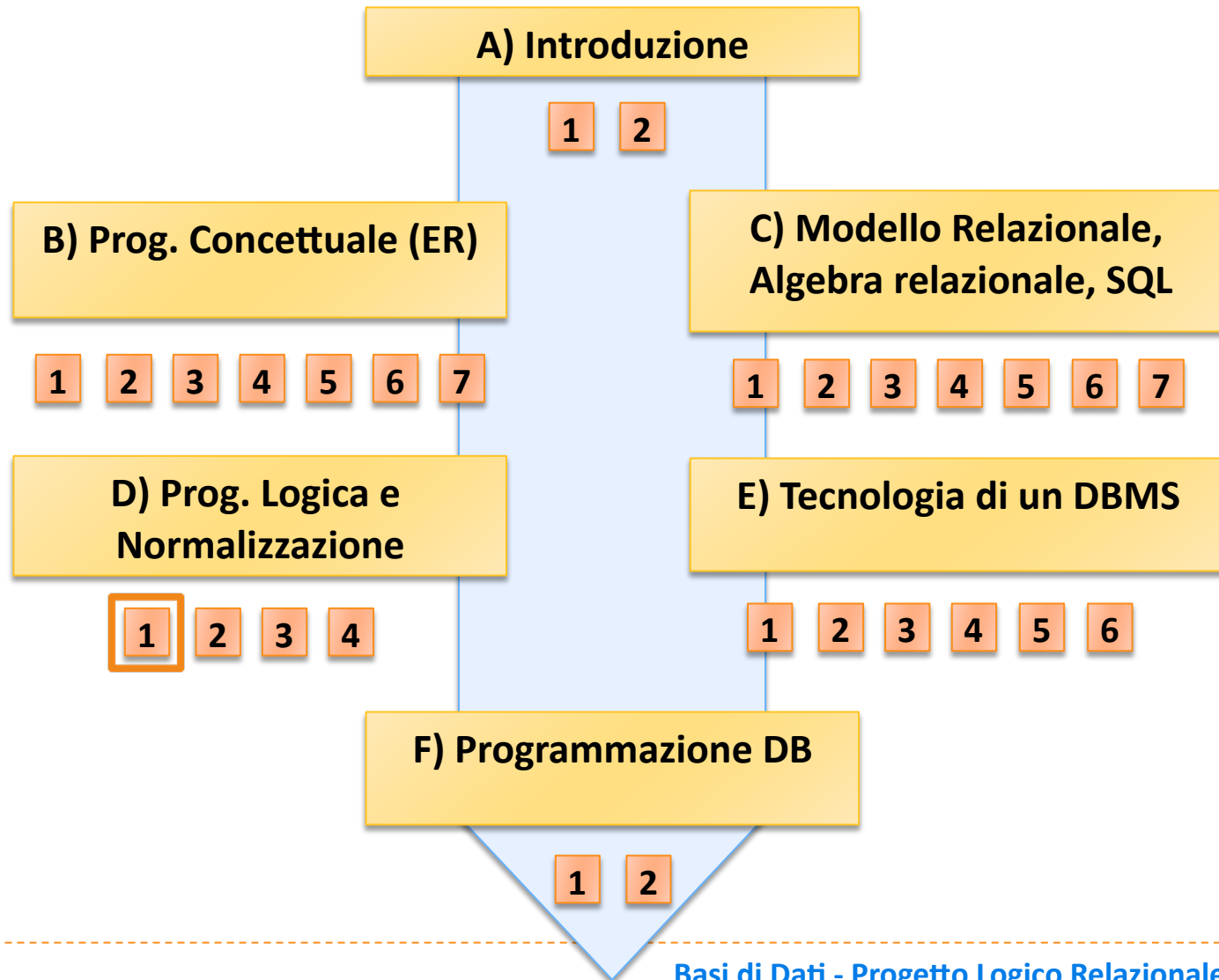


Basi di Dati

Progetto Logico Relazionale (Parte 1)

Basi di Dati – Dove ci troviamo?



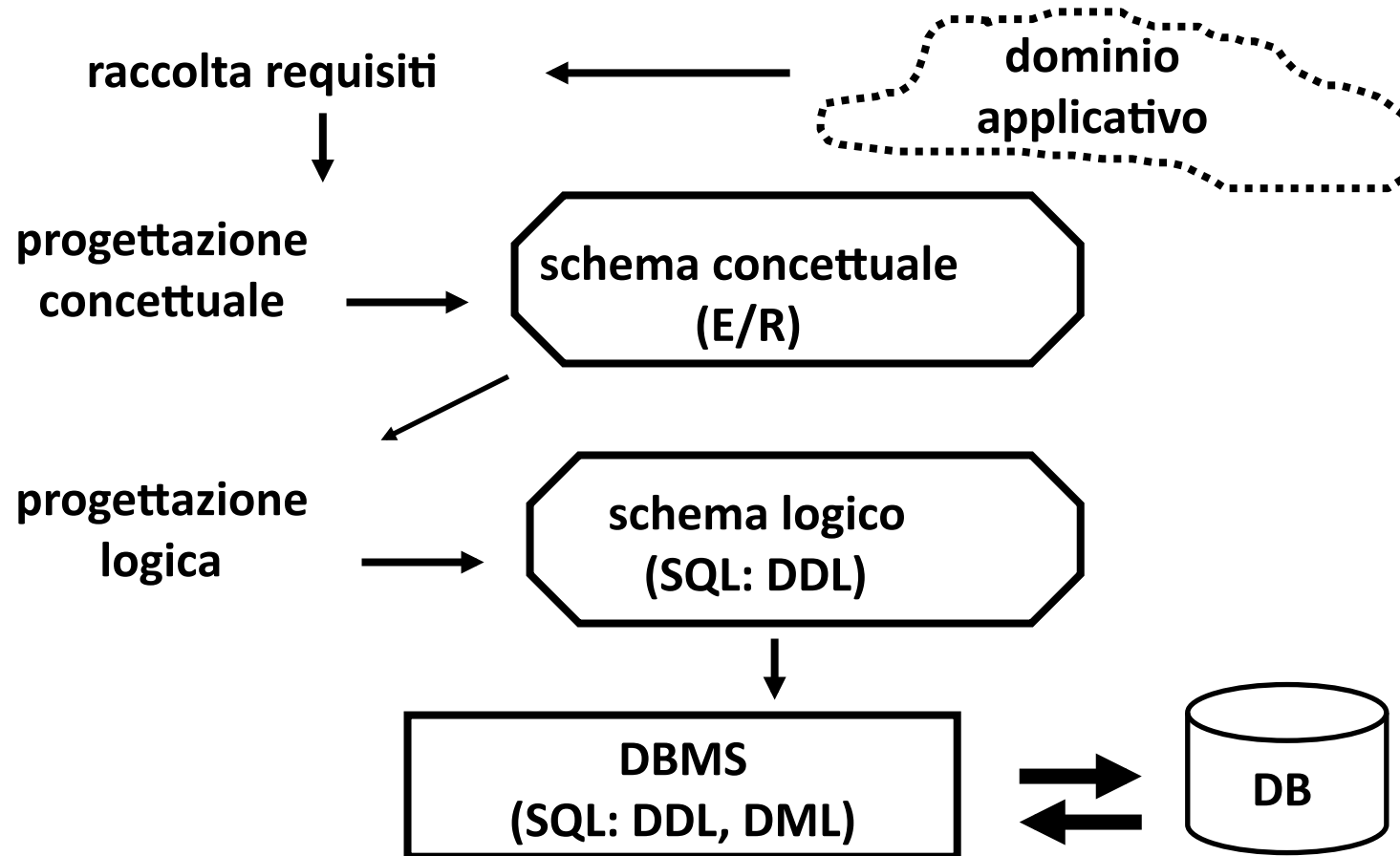
Progetto logico

- ▶ Lo schema **E/R** descrive un **dominio applicativo** ad un dato livello di astrazione
- ▶ Lo schema E/R è molto utile per:
 - ▶ fornire una **descrizione** sintetica e visiva
 - ▶ rappresentare buona parte della **semantica** dell'applicazione
 - ▶ scambiare **informazioni** sull'attività progettuale tra i membri del team di progetto e mantenere una **documentazione**

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



Scelte alternative

- il progetto logico presenta in generale una **soluzione standard** determinabile in modo semplice e diretto
- a seconda dei casi sono però disponibili anche **soluzioni alternative** che possono rivelarsi più o meno convenienti
- per una scelta corretta sono necessarie:
 - previsioni sulla natura e la frequenza delle **operazioni**
 - valutazioni quantitative sui **volumi di dati** (entità, associazioni, percentuali di copertura di gerarchie, percentuali di valori nulli)

Scelte alternative

si possono individuare alcune regole intuitive:

- ▶ le **proprietà logiche** sono comunque primarie rispetto ai motivi di efficienza
- ▶ tenere sulla **stessa entità** informazioni che verranno di frequente consultate insieme
- ▶ tenere su **entità separate** informazioni che verranno consultate separatamente
- ▶ limitare l'incidenza di **valori nulli** per attributi opzionali

Fasi del progetto

il progetto produce trasformazioni e traduzioni dello schema E/R con le seguenti fasi:

- 1 **eliminazione delle gerarchie isa**
- 2 **selezione delle chiavi primarie, eliminazione delle identificazioni esterne**
- 3 **trasformazione degli attributi composti o multipli**
- 4 **traduzione di entità e associazioni in schemi di relazioni**
- 5 **verifica di normalizzazione**

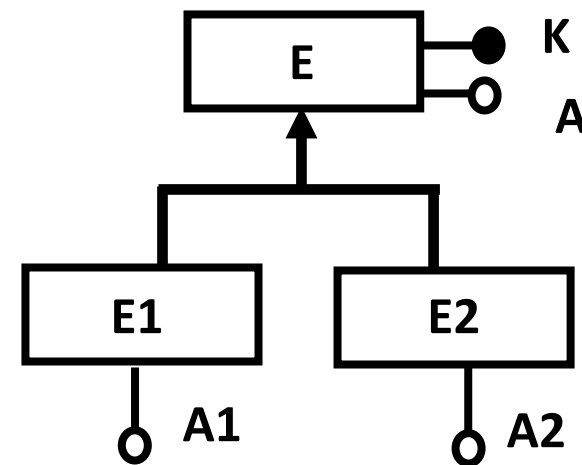
Fasi del progetto

- ▶ dopo le prime tre fasi, lo schema E/R è costituito soltanto da entità, associazioni e attributi semplici
 - ▶ la quarta fase non è complessa
 - ▶ la quinta richiede molta attenzione
- ! si tenga presente che ogni trasformazione **impoverisce** lo schema; la semantica persa deve restare sotto forma di **vincoli di integrità** che governeranno l'utilizzo delle relazioni

Eliminazione delle gerarchie

il modello relazionale non rappresenta le gerarchie, le gerarchie sono sostituite da entità e associazioni:

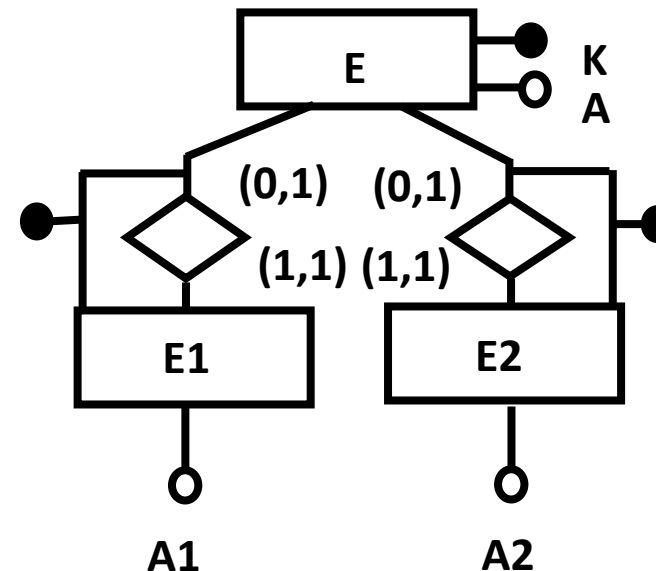
- 1) mantenimento delle entità con associazioni
- 2) collasso verso l'alto
- 3) collasso verso il basso



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

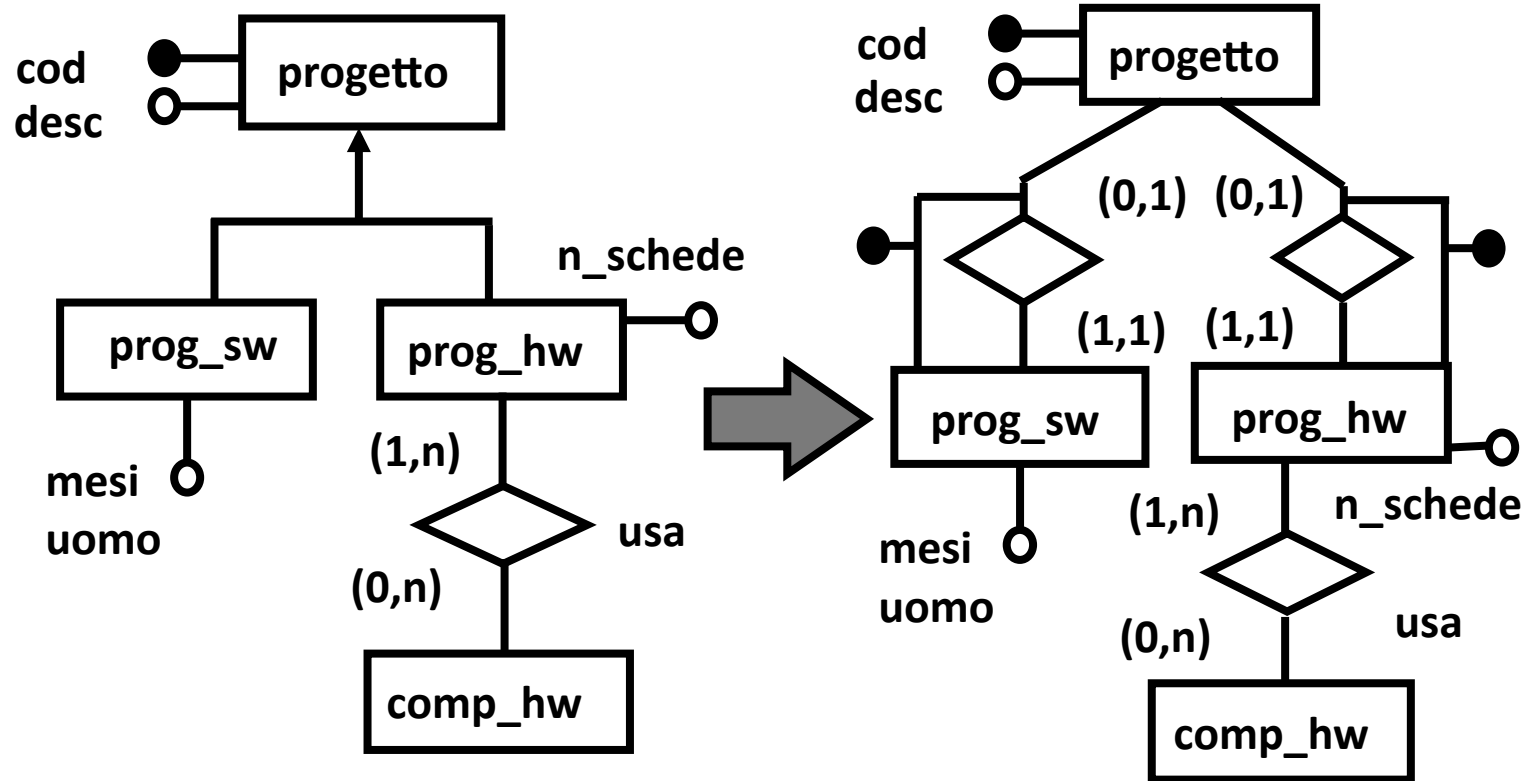
Mantenimento delle entità

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



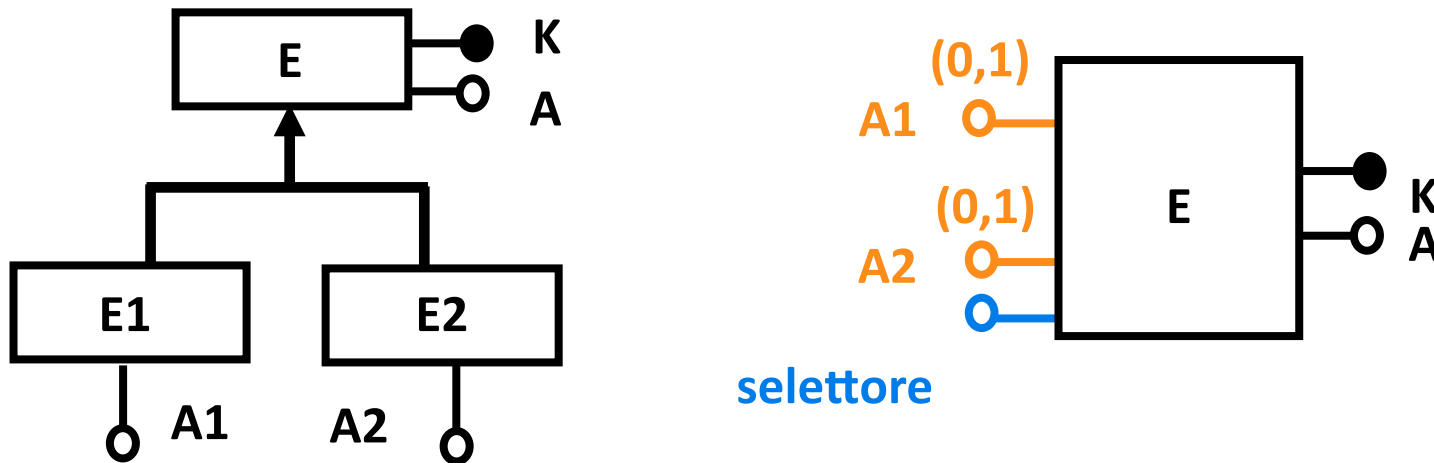
questa soluzione è sempre possibile,
indipendentemente dalla copertura

Mantenimento entità - es.:



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,

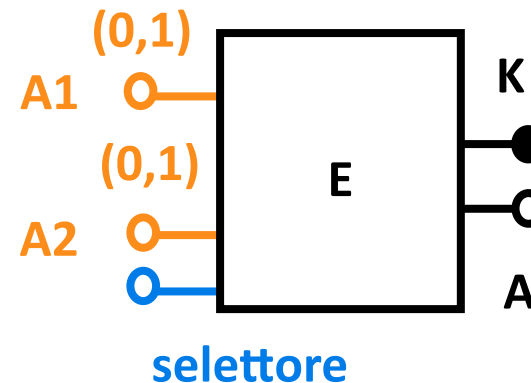
quante sono le sottoentità

parziale esclusiva:

selettore ha **N+1** valori; il valore in più serve

per le istanze che non appartengono ad

alcuna sottoentità

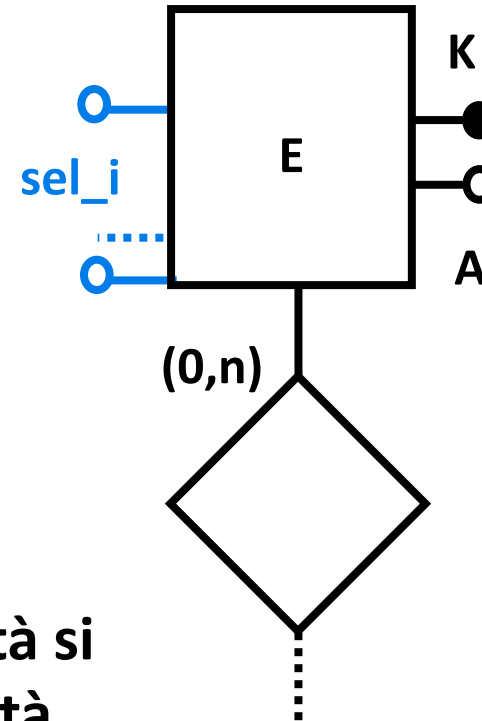


ISA: collasso verso l'alto

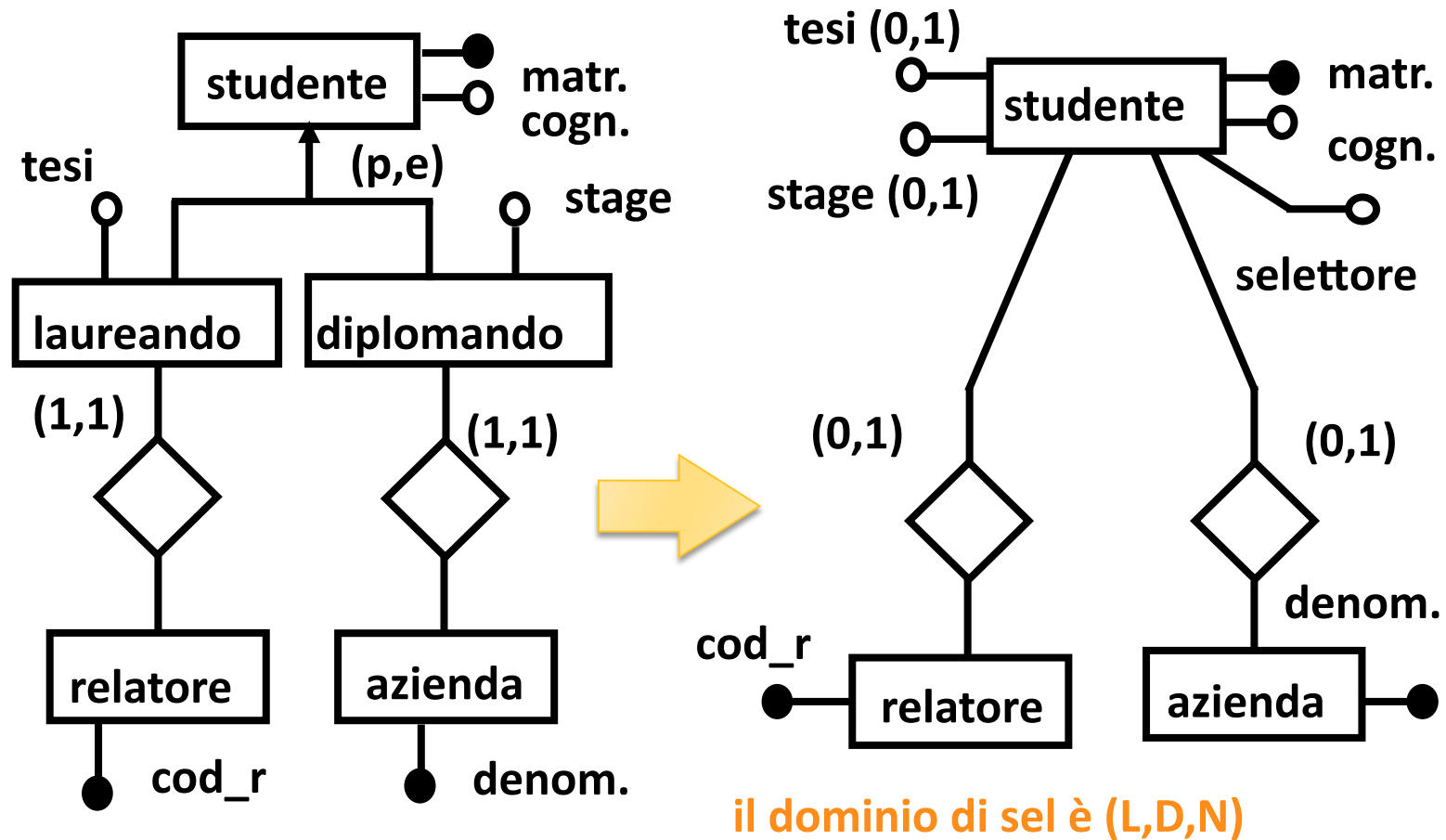
Copertura dell'ISA:

non esclusiva: 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 l'alto

studente(123, rossi)
studente(218, bianchi)
studente(312, verdi)

laureando(123, DFD)

diplomando(312, turbina)



(selettore)



studente(123,rossi, L, DFD, NULL)
studente(218,bianchi, N, NULL, NULL)
studente(312,verdi, D, NULL, turbina)

ISA: collasso verso l'alto

- ▶ **esiste una precisa relazione tra il valore del selettore e i campi che possono avere valore diverso da NULL**
- ▶ **campi prima obbligatori ora ammettono il valore NULL**
- ▶ **per una stima delle percentuali di NULL occorre conoscere le percentuali di laureandi e diplomandi**

ISA: collasso verso il basso

Collasso verso il basso:

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

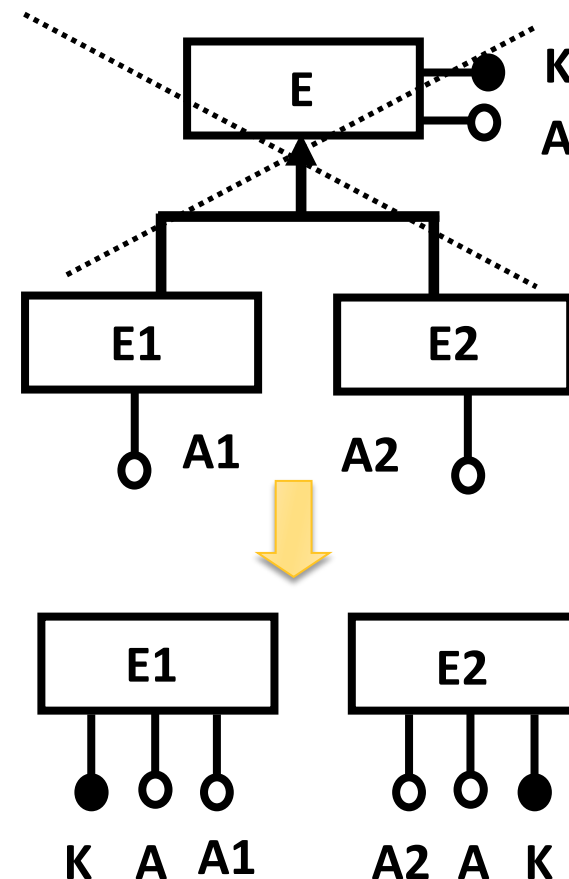
ISA: collasso verso il basso

limiti di applicabilità:

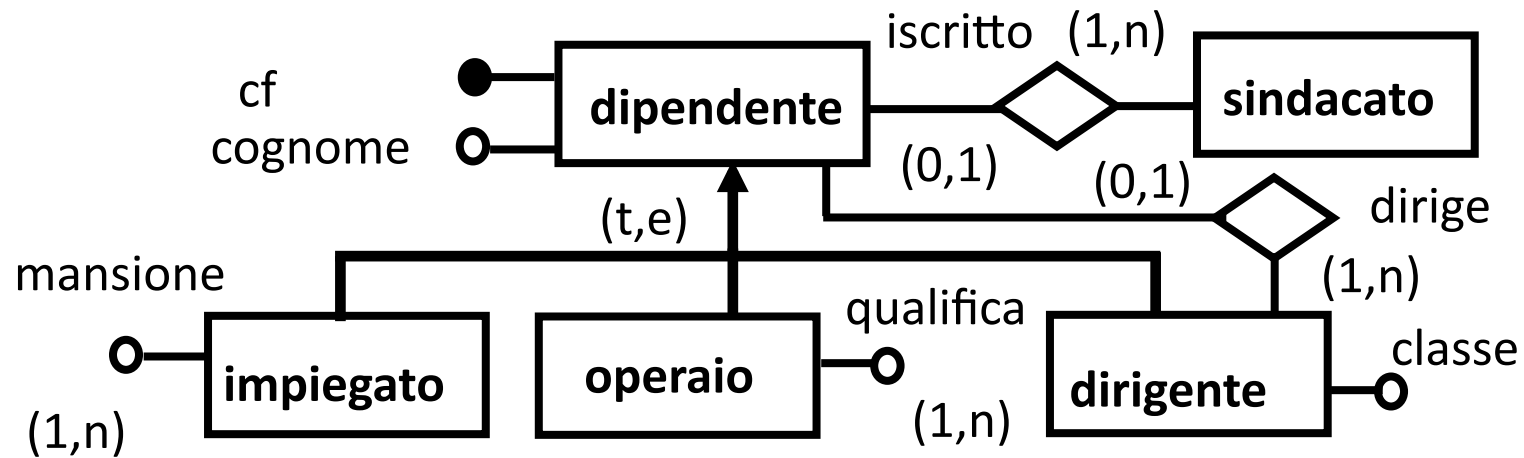
- se la copertura non è totale
non si può fare:

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

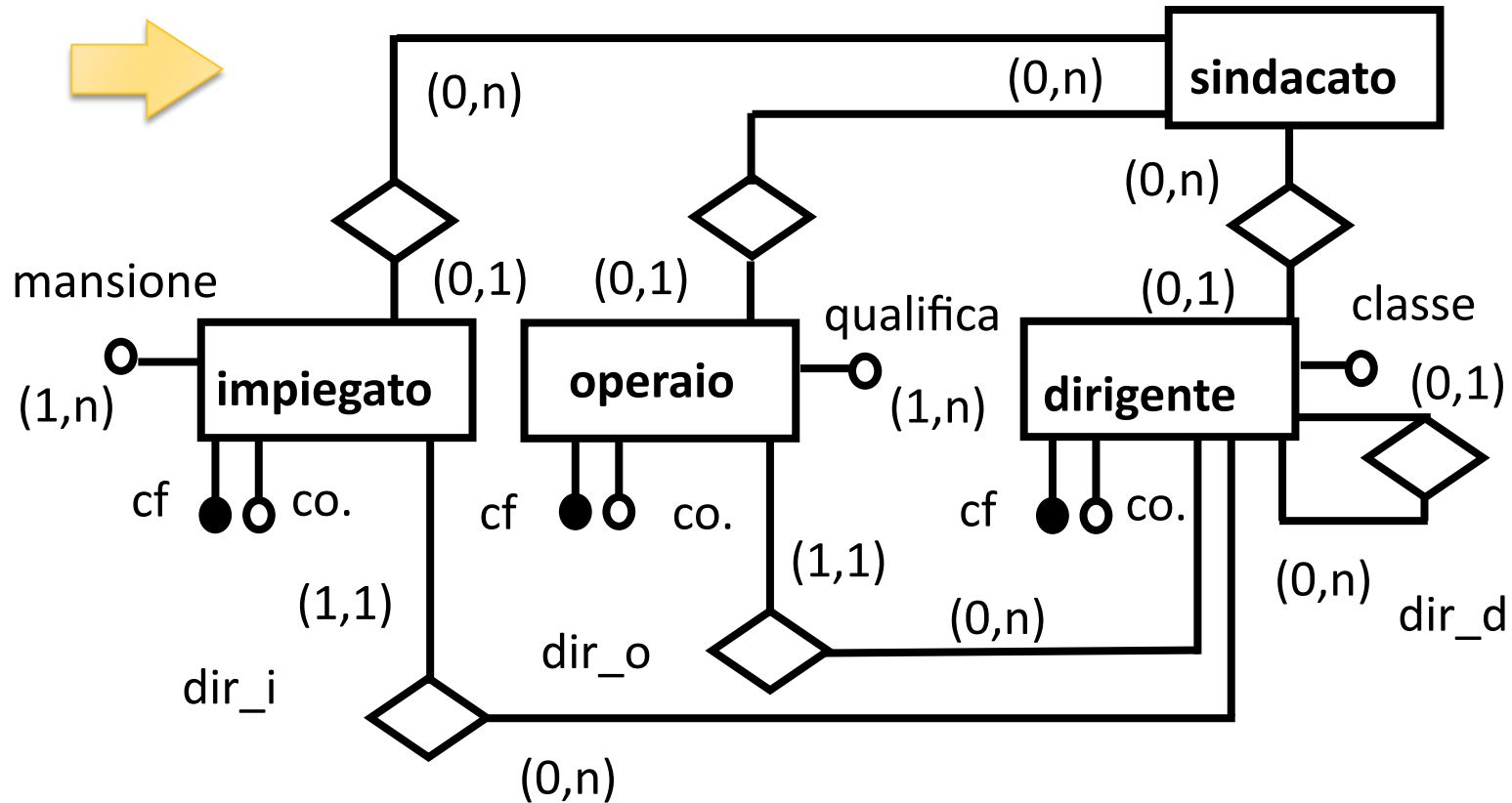
- se la copertura non è esclusiva **introduce ridondanza:** per una istanza presente sia in E1 che in E2 si rappresentano due volte gli attributi di E



Collasso verso il basso: es.



Collasso verso il basso: es.

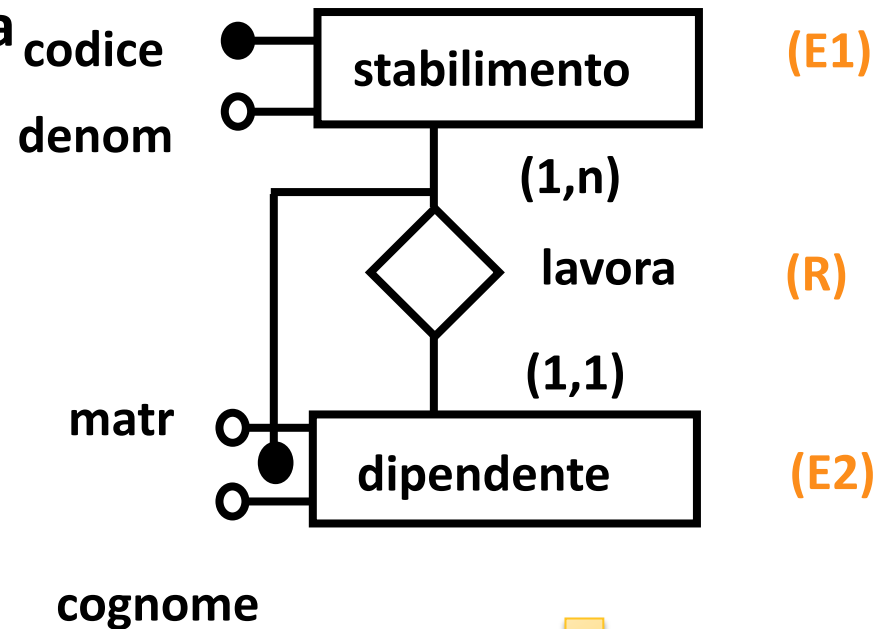


Scelta della chiave primaria

- ▶ È necessario che tra i diversi **identificatori** di una entità venga designata una **chiave primaria**: per la **chiave primaria** occorrerà, infatti, che il DBMS sia provvisto di strumenti per garantire **l'unicità dei valori**
- ▶ **criteri euristici di scelta**:
 - ▶ **primo**: scegliere la chiave che è usata più frequentemente per accedere all'entità
 - ▶ **secondo**: si preferiscono chiavi semplici a chiavi composte, interne anziché esterne

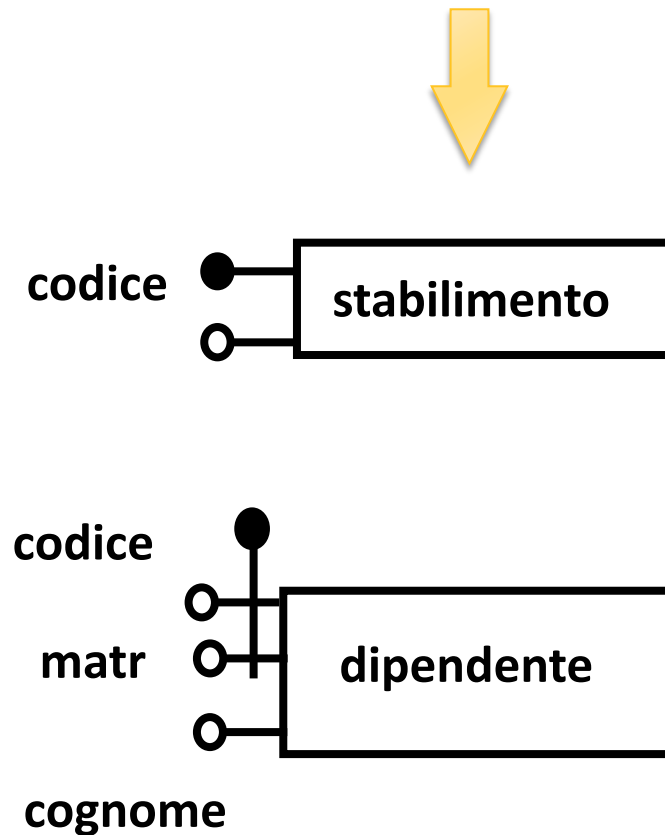
Identificatori esterni

- ▶ 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



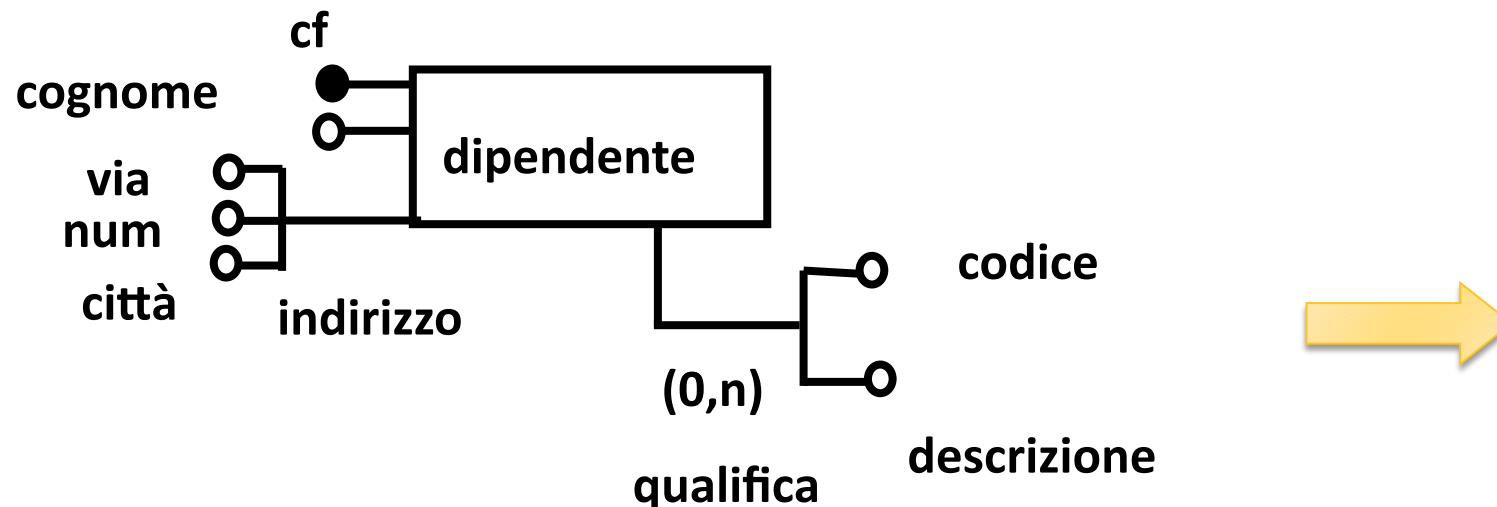
Identificatori esterni

- ▶ 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



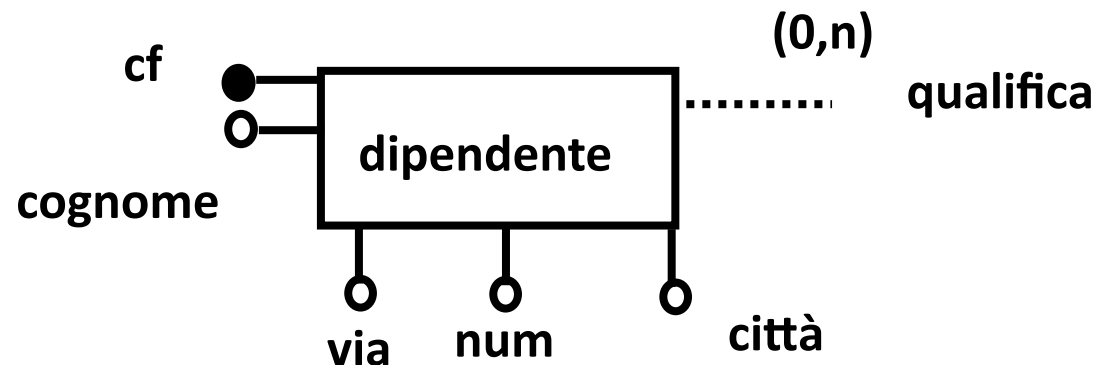
Attributi composti/ripetuti

le relazioni **non possono** (per definizione) contenere attributi **composti** o, attributi **ripetuti** ma solamente attributi **“atomici”**



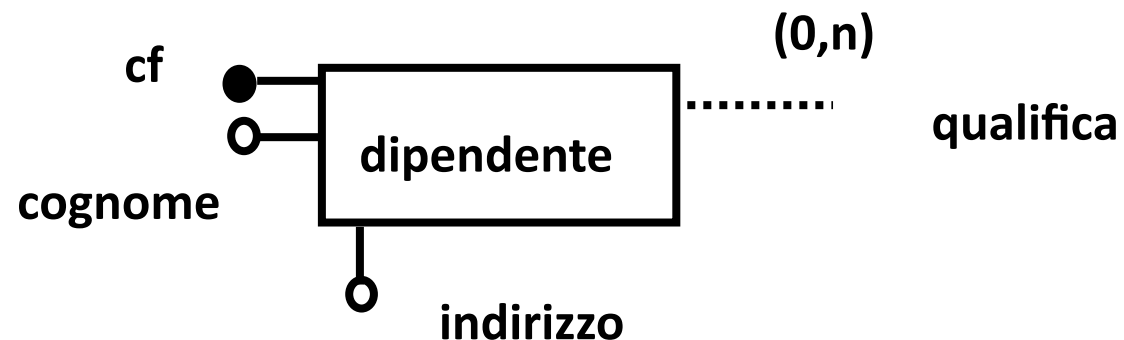
Attributi composti

- ▶ **Due possibili soluzioni**
 - ▶ **eliminare l'attributo composto** e considerare i suoi componenti come attributi semplici
 - ▶ in questo modo si perde la visione unitaria ma si mantiene l'articolazione dei componenti



Attributi composti

- ▶ **eliminare i componenti** e considerare l'attributo come semplice
 - ▶ in questo modo lo schema risulta semplificato, perdendo parte dei dettagli

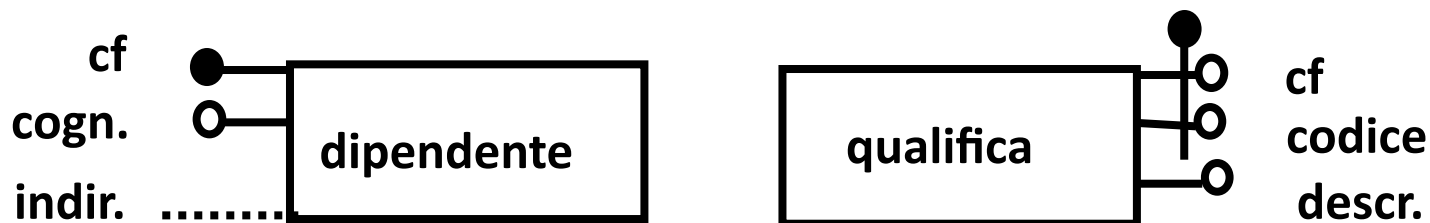


Attributi ripetuti

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:

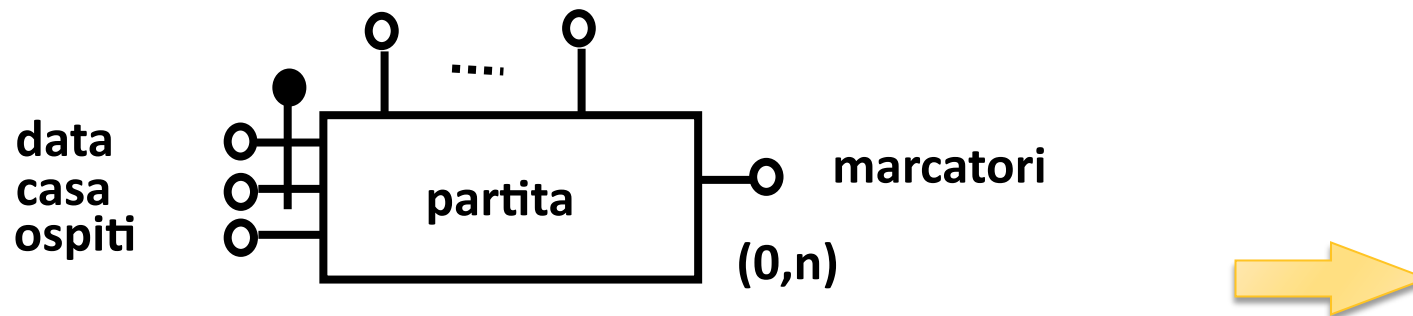
la nuova entità EA ha l'identificatore **composto** dall'identificatore di E più l'attributo A



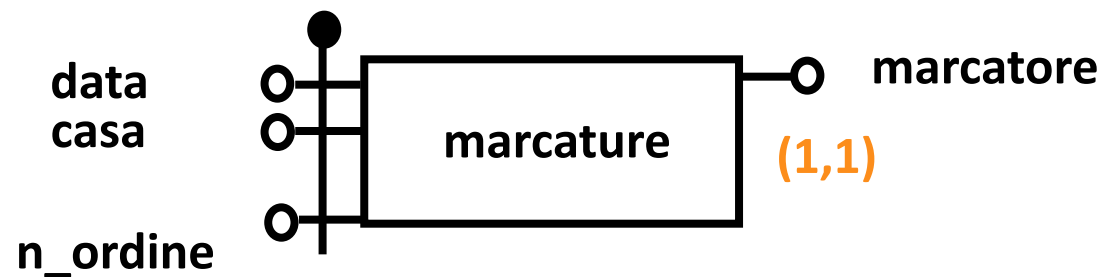
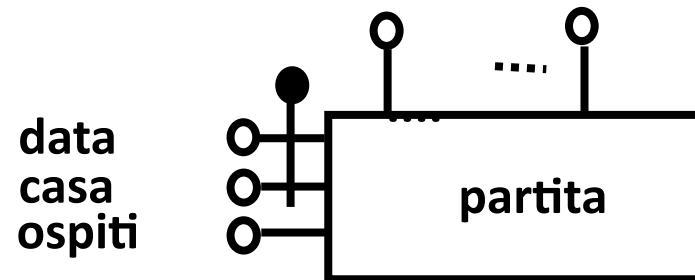
Attributi ripetuti

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)



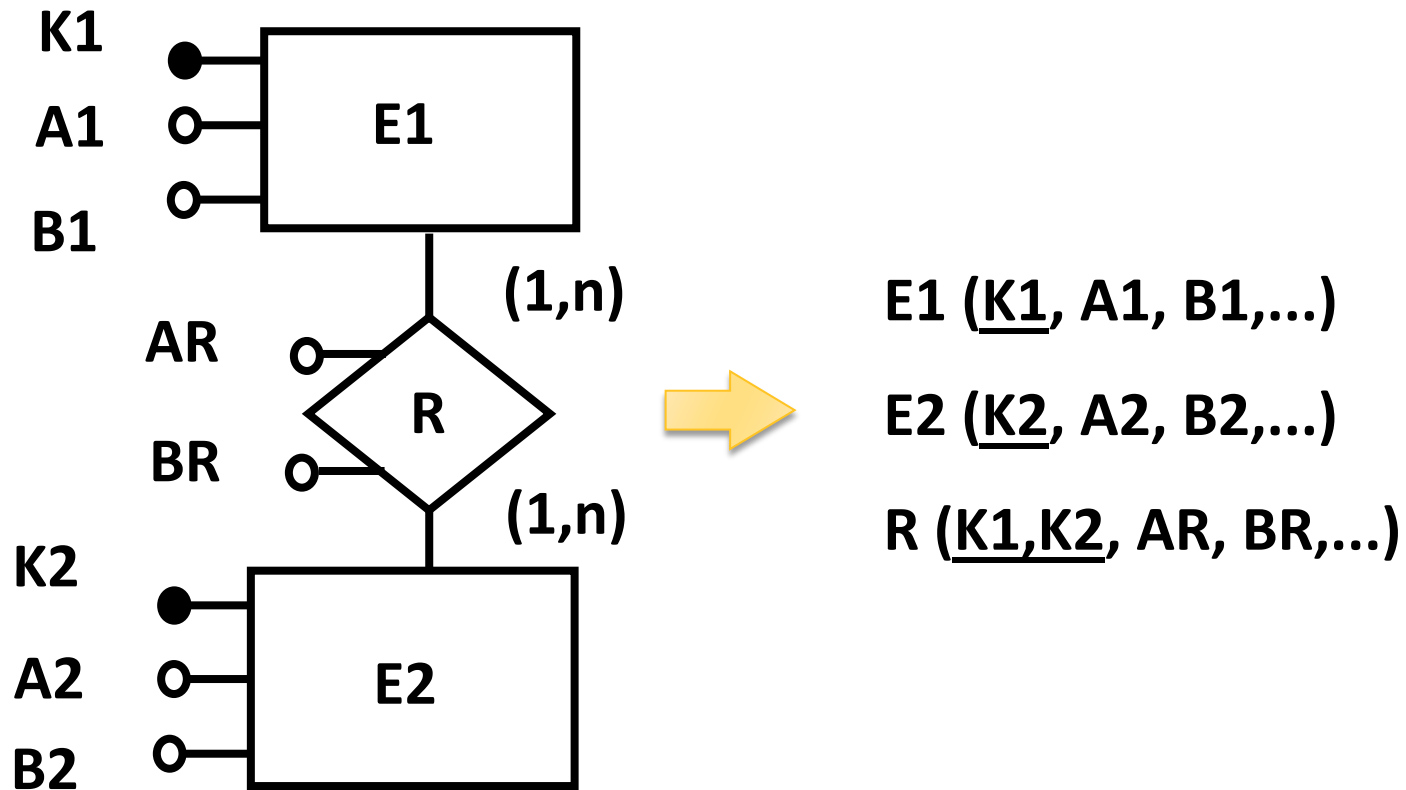
Attributi ripetuti



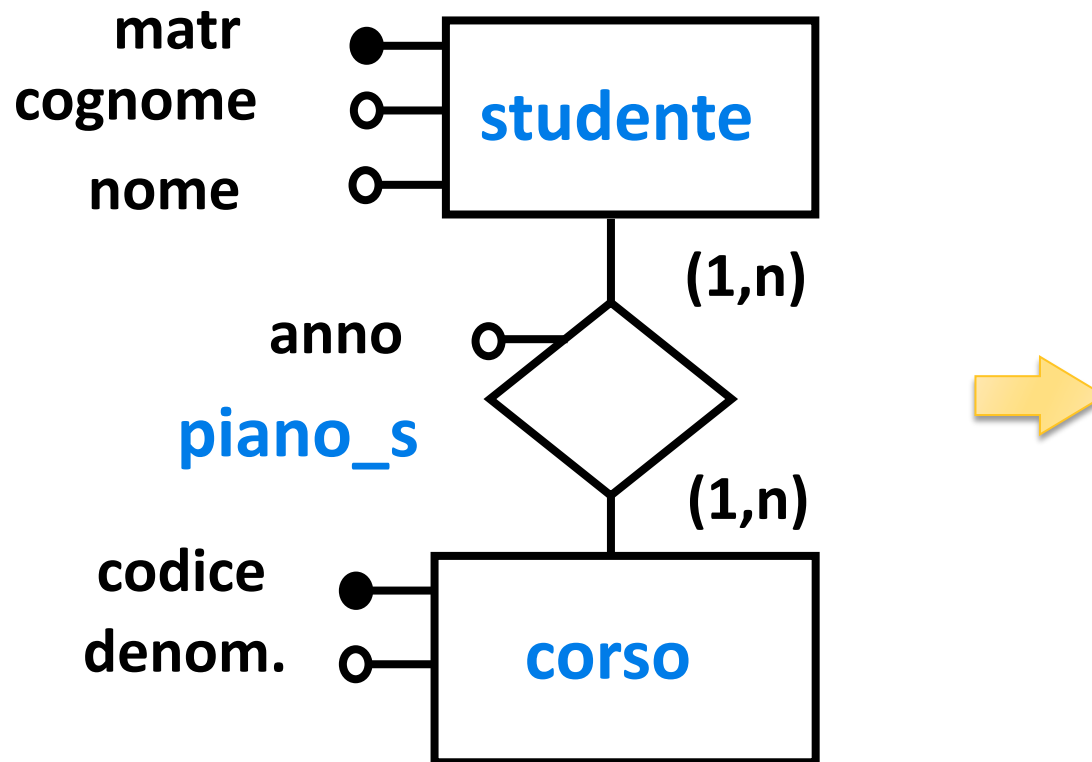
Traduzione standard

- ▶ ogni **entità** è tradotta con **una relazione** con gli stessi attributi
 - ▶ la **chiave** è la chiave (o identificatore) dell'entità stessa (**già visto**)
- ▶ ogni **associazione** è tradotta con **una relazione** con gli stessi attributi, cui si aggiungono gli identificatori di tutte le entità che essa collega (**già visto**)
 - ▶ la **chiave** è composta dalle chiavi delle entità collegate (questa può però essere una **superchiave**, come si vedrà in seguito)

Traduzione standard



Traduzione standard es.



Traduzione standard es.

STUDENTE (MATR, COGNOME, NOME)

CORSO (CODICE, DENOM)

PIANO_ST (MATR, CODICE, ANNO)

FK: MATR **REFERENCES** STUDENTE

FK: CODICE **REFERENCES** CORSO



in SQL...

Traduzione standard es.

```
CREATE TABLE STUDENTE (MATR... NOT NULL,  
..., NOME... , PRIMARY KEY (MATR));
```

```
CREATE TABLE CORSO (CODICE... NOT NULL,  
DENOM ... , PRIMARY KEY (CODICE));
```

```
CREATE TABLE PIANO_ST (MATR... NOT NULL,  
CODICE... NOT NULL, ANNO...  
PRIMARY KEY (MATR, CODICE),  
FOREIGN KEY (MATR) REFERENCES STUDENTE  
FOREIGN KEY (CODICE) REFERENCES CORSO);
```