

# Progettazione logica



**DOCENTE**  
**PROF. ALBERTO BELUSSI**

**Anno accademico 2011/12**

# Riepilogo lezioni precedenti

2

- **Modello relazionale**

- **Costrutti introdotti**

- ✦ Domini di base: caratteri, stringhe di caratteri, interi, reali, domini del tempo
- ✦ Relazione (o tabella)
  - Definizione come insieme di *ennuple*
  - Definizione come insieme di *tuple*
- ✦ Chiavi primarie
- ✦ Vincoli di integrità referenziale

- **Schema relazionale di una base di dati:**

- ✦  $R_1(A_{1,1}, \dots, A_{1,n_1}), \dots, R_k(A_{k,1}, \dots, A_{k,n_k})$

# Progettazione logica

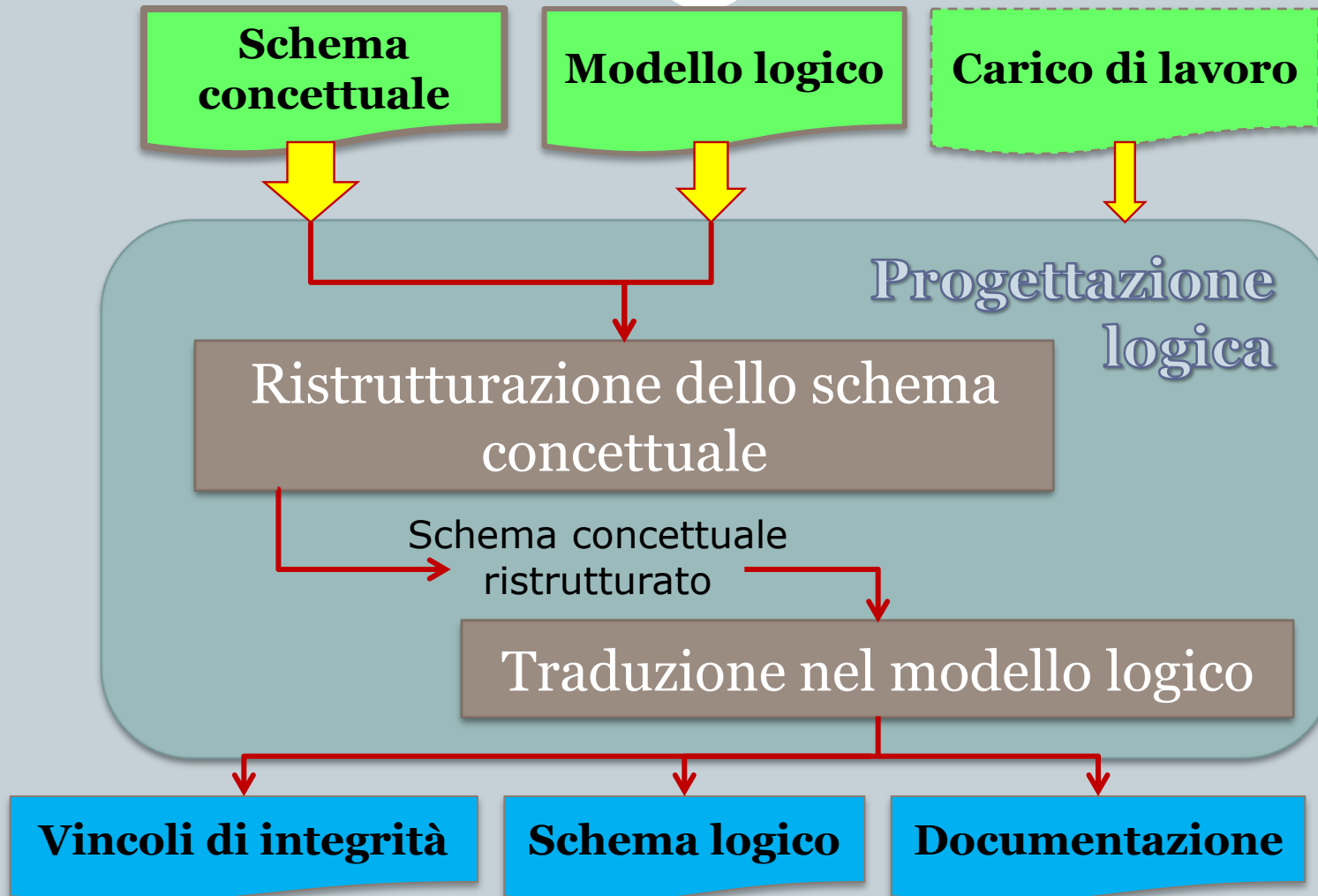
3

## **Obiettivo della progettazione logica**

**Produrre uno schema logico che descriva in modo CORRETTO ed EFFICACE tutte le informazioni contenute nello schema concettuale**

# Progettazione logica

4



# Progettazione logica

5

## Motivazioni della ristrutturazione

- Semplificare la fase successiva di traduzione
- Ottimizzare le strutture rispetto al carico di lavoro

## Carico di lavoro

- Dimensione dei dati
- Caratteristiche delle operazioni:
  - ✦ Costo
  - ✦ Frequenza

# Ristrutturazione dello schema ER

6

## **Sottofasi:**

- **Analisi delle ridondanze dovute a presenza di dati derivabili**
- **Eliminazione delle generalizzazioni**
- **Accorpamento/partizionamento di entità e relazioni**
- **Scelta degli identificatori principali**

# Analisi delle derivabilità

7

## **Derivabilità**

Presenza di dati derivabili da altri dati.

- Attributi derivabili da altri attributi della stessa entità (o relazione)
- Attributi derivabili da attributi di altre entità (o relazioni)
- Attributi derivabili dal conteggio di occorrenze
- Relazioni derivabili dalla composizione di altre relazioni

# Analisi delle ridondanze

8

## Obiettivo

Analizzare i dati derivabili presenti nello schema (o richiesti dalle operazioni) e decidere se:

- A. Mantenere il dato derivabile
- B. Calcolare il dato derivabile quando serve

(A) conviene quando:

$$\text{COSTO\_Calcolo} > \text{COSTO\_Aggiornamento}$$

(B) conviene quando:

$$\text{COSTO\_Calcolo} < \text{COSTO\_Aggiornamento}$$

Il COSTO viene misurato in #accessi/unità di tempo in quanto vengono considerate le frequenze di calcolo e di aggiornamento del dato



# Eliminazione delle generalizzazioni

9

## Obiettivo

Sostituire le **generalizzazioni** presenti nello schema ER con strutture alternative, in quanto non direttamente rappresentabili nel modello relazionale.

Esistono tre soluzioni alternative:

- Accorpamento delle entità figlie nel padre
- Accorpamento del padre nelle entità figlie
- Sostituzione della generalizzazione con relazioni

# Eliminazione delle generalizzazioni

10

## Soluzione 1 – accorpamento nel padre

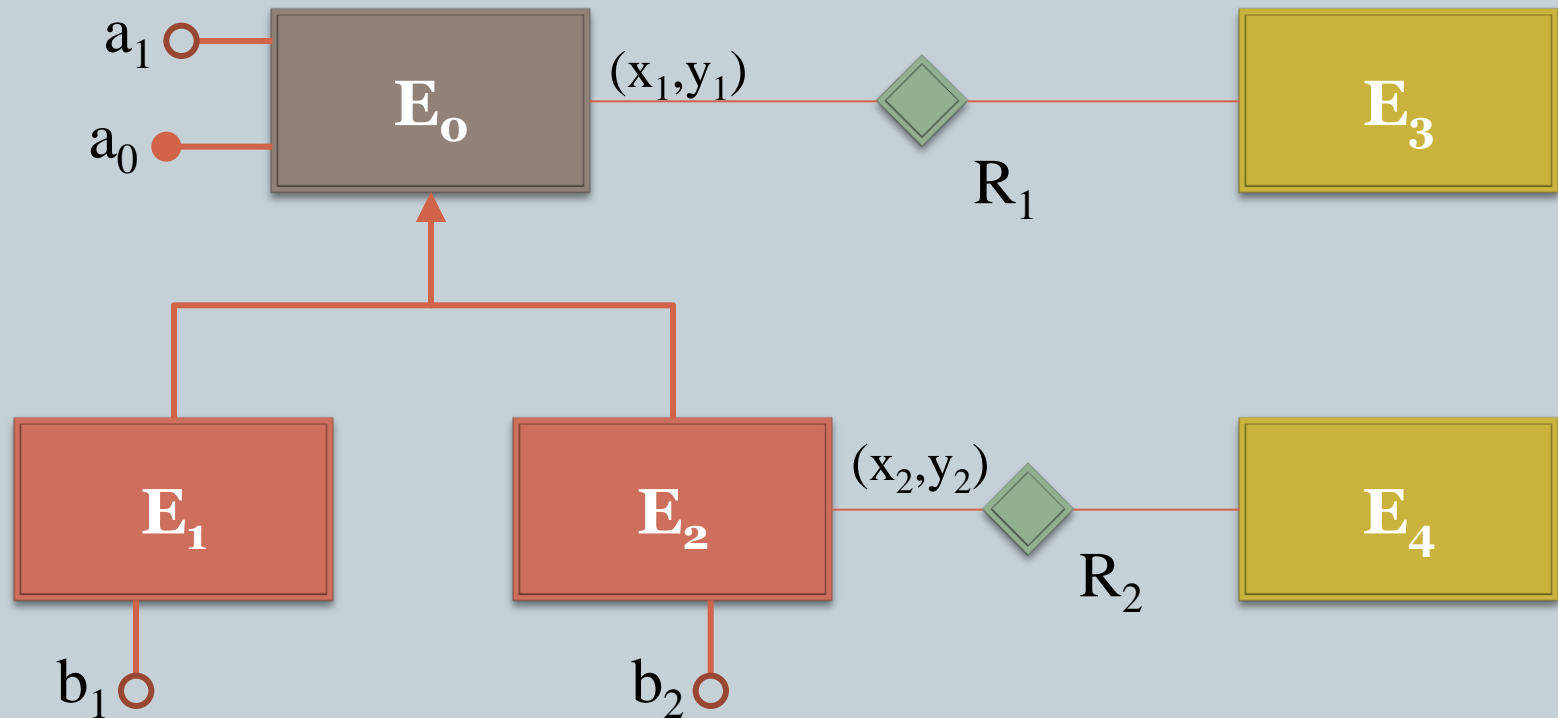
### Passi:

- Eliminazione delle entità figlie.
- Accorpamento nel padre di tutti gli attributi specifici delle entità figlie come attributi opzionali.
- Accorpamento nel padre delle relazioni che coinvolgono le entità figlie assegnando cardinalità minima uguale a zero (lato entità padre).
- Aggiunta di un attributo TIPO per distinguere nel padre le occorrenze delle entità figlie eliminate.

# Accorpamento nel padre

11

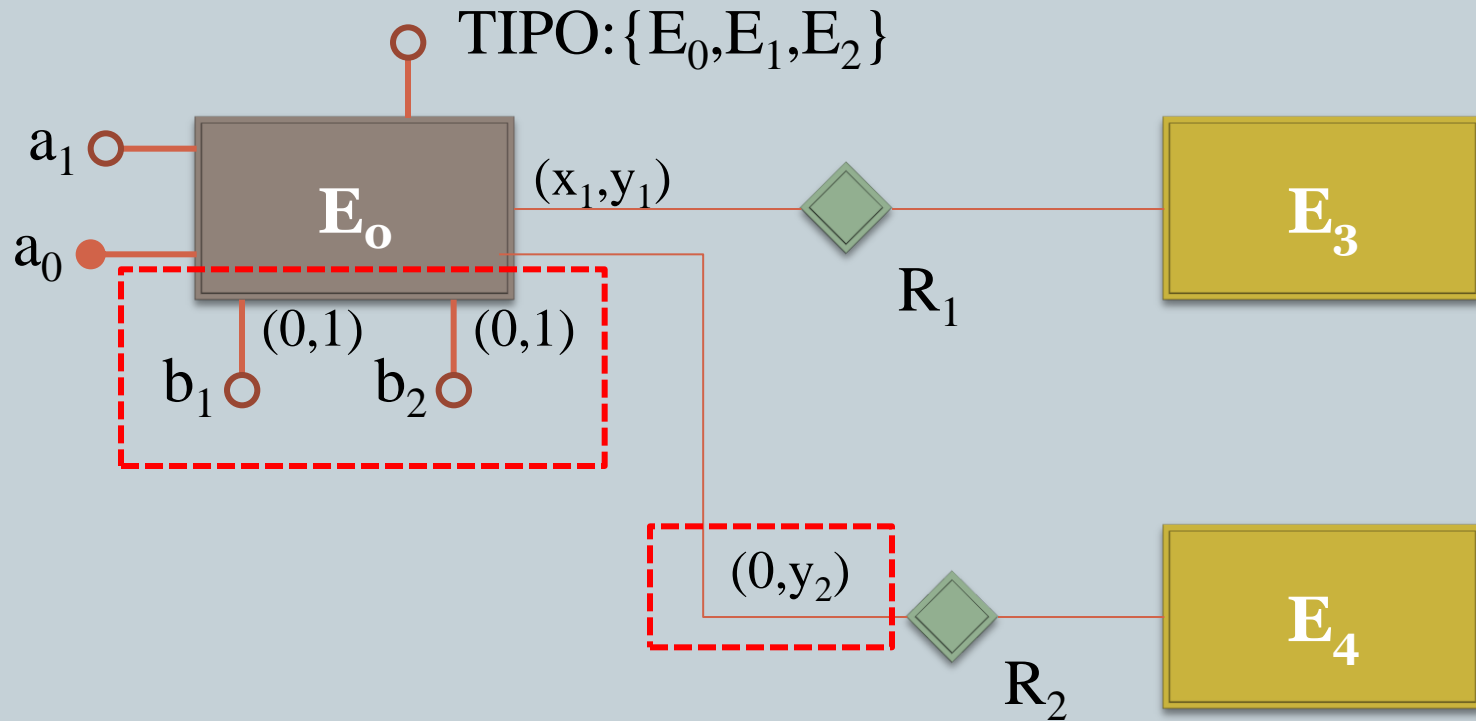
Esempio: stato iniziale



# Accorpamento nel padre

12

Esempio: dopo l'accorpamento



# Eliminazione delle generalizzazioni

13

## Soluzione 2 – accorpamento nei figli

### Precodizione:

La generalizzazione deve essere TOTALE.

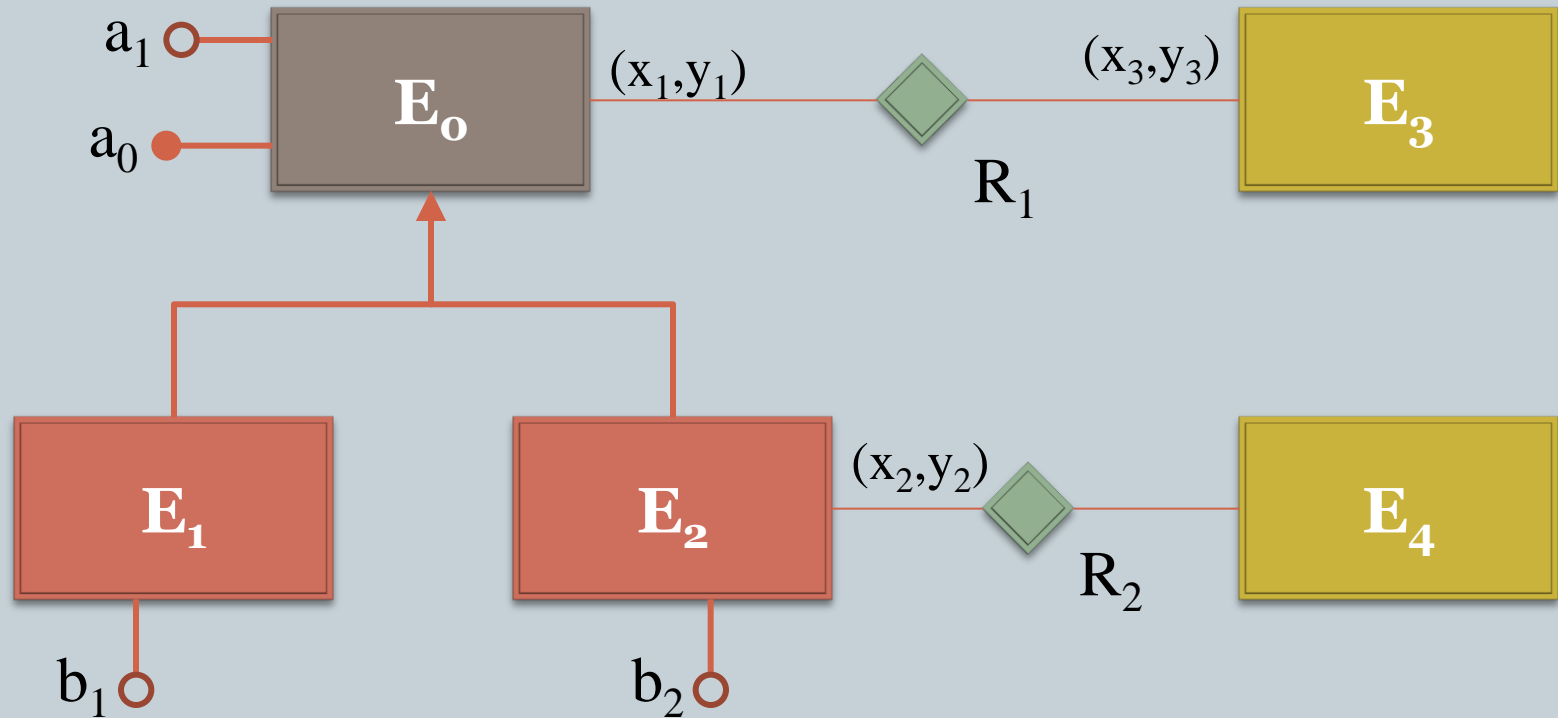
### Passi:

- Eliminazione dell'entità padre.
- Replicazione degli attributi dell'entità padre su ogni entità figlia.
- Partizionamento sui figli delle relazioni che coinvolgono l'entità padre assegnando cardinalità minima uguale a zero (lato entità esterne).

# Accorpamento nei figli

14

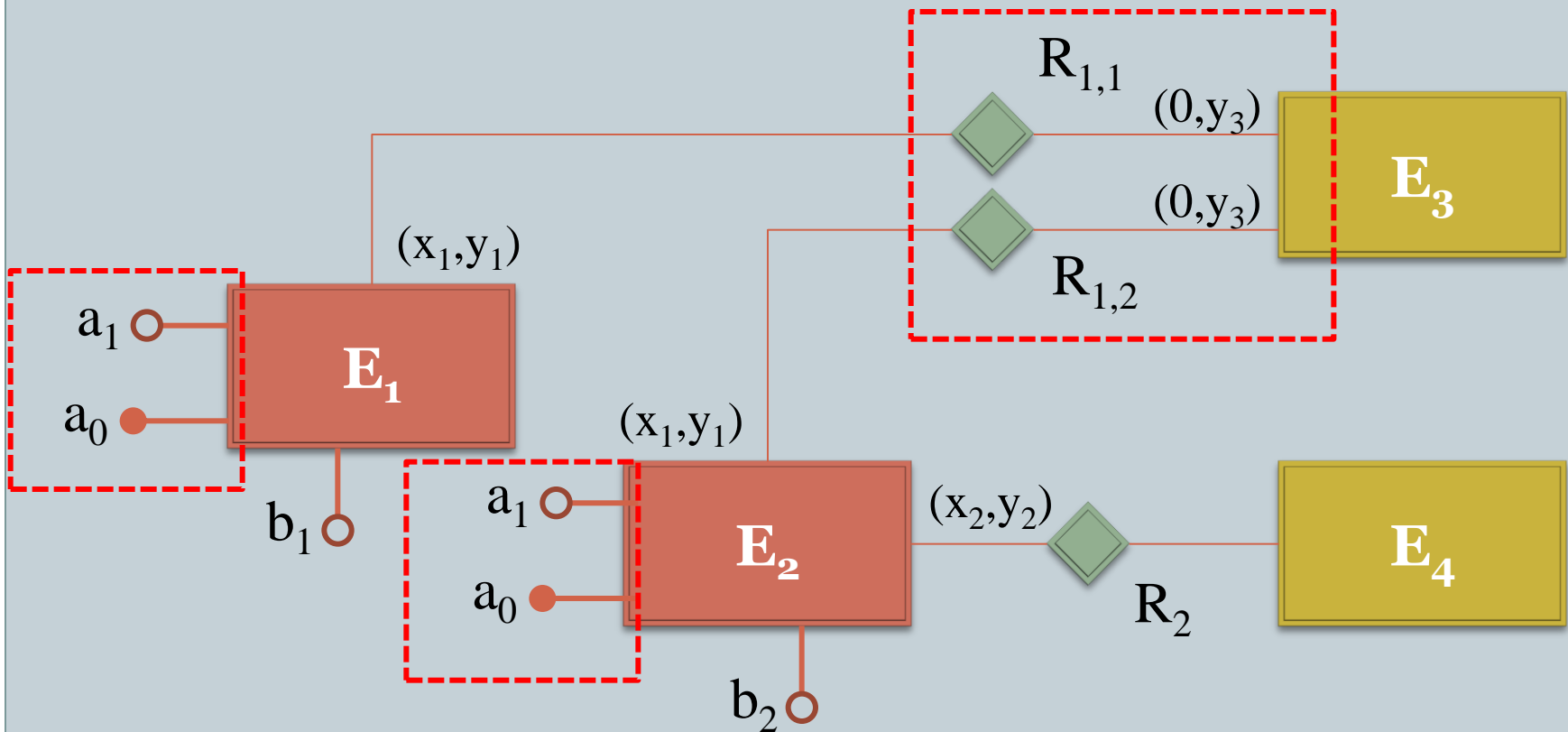
Esempio: stato iniziale



# Accorpamento nei figli

15

Esempio: dopo l'accorpamento nei figli



# Eliminazione delle generalizzazioni

16

## Soluzione 3 – sostituzione con relazioni

### Passi:

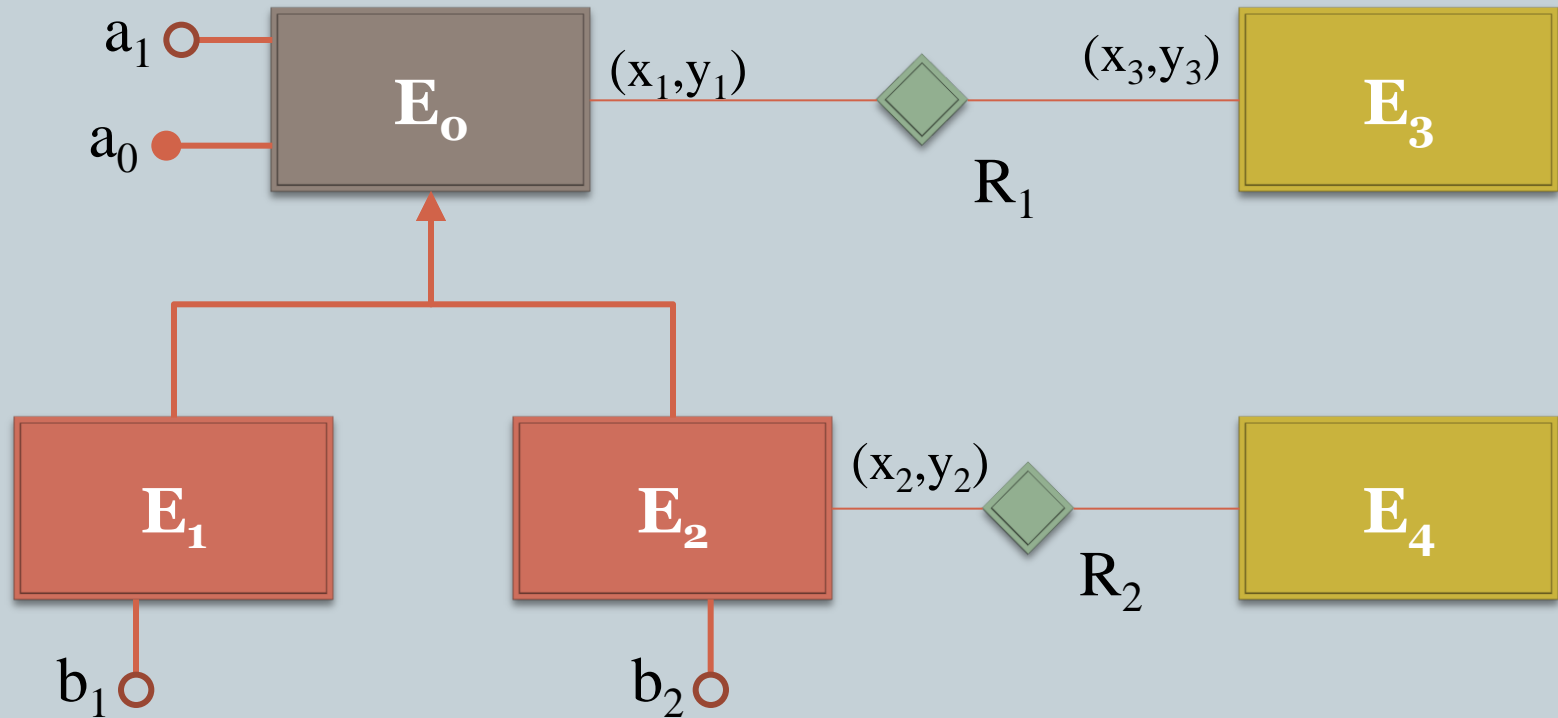
- Eliminazione della generalizzazione.
- Inserire n relazioni tra il padre e ciascuna delle n entità figlie.
- Tutti gli attributi conservano la loro collocazione.



# Sostituzione con relazioni

17

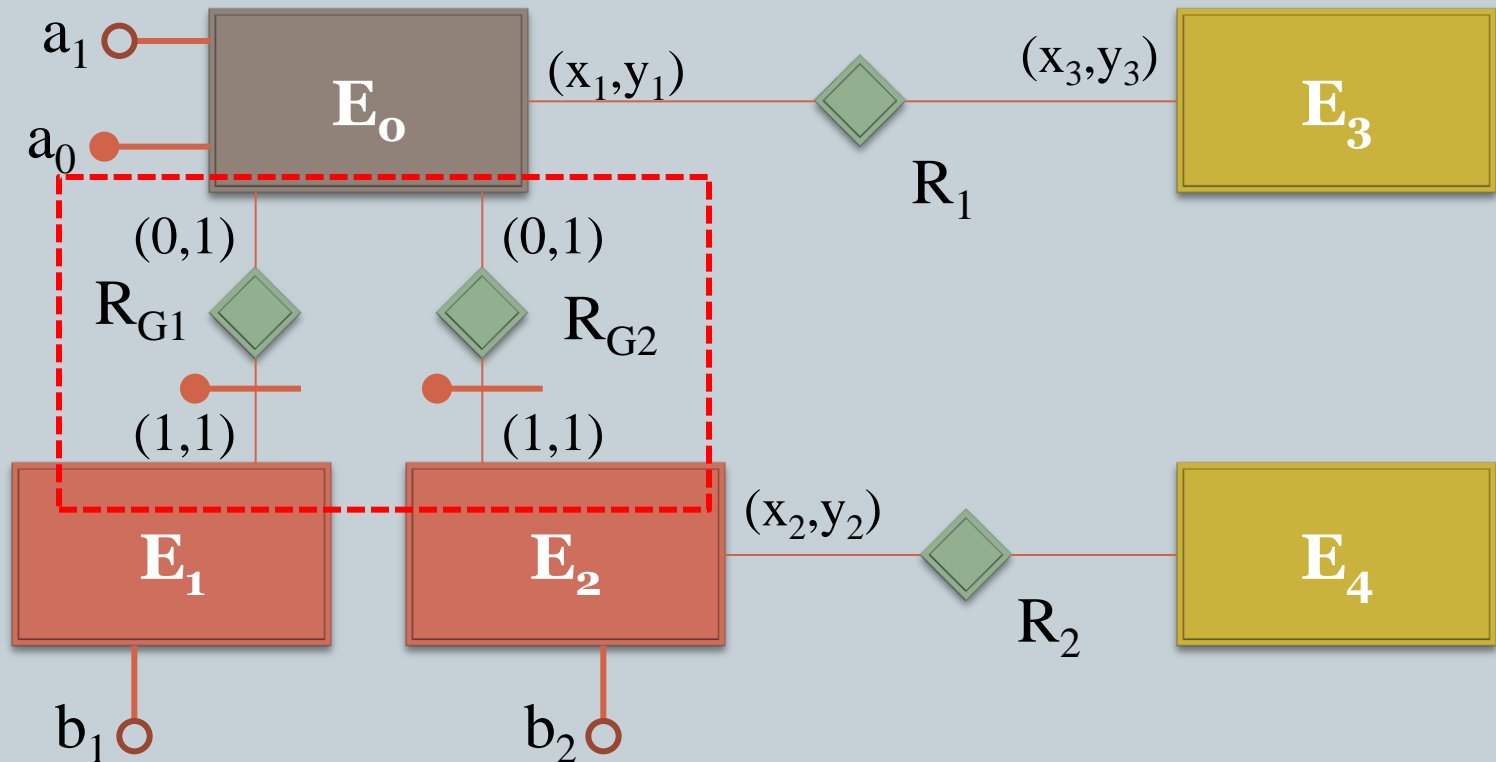
Esempio: stato iniziale



# Sostituzione con relazioni

18

Esempio: stato iniziale

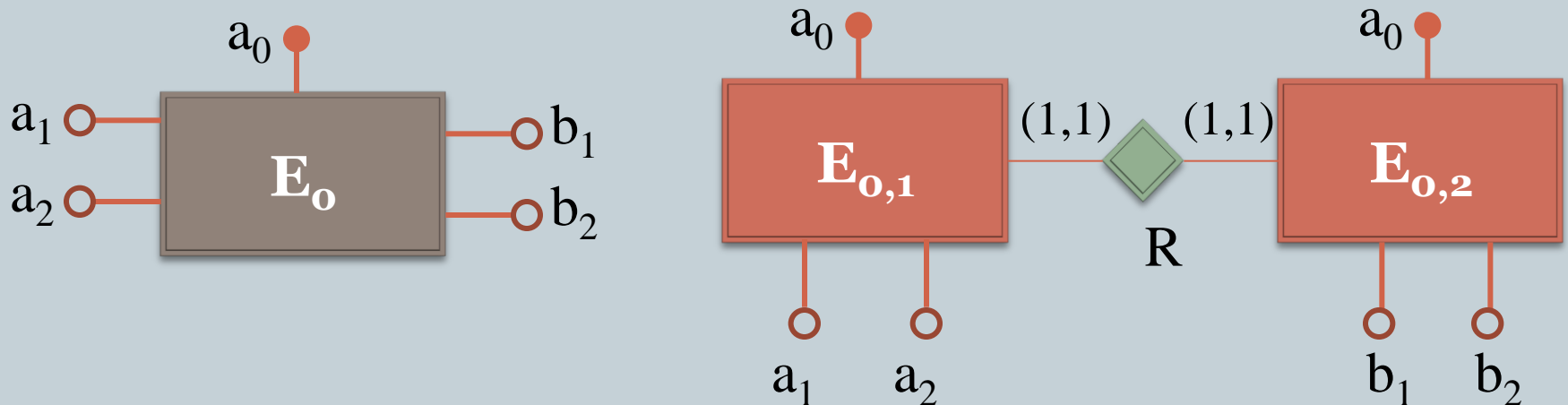


# Partizionamento di entità

19

Sostituire un'entità dello schema con una coppia di entità con lo stesso identificatore e una relazione uno a uno che le lega tra loro.

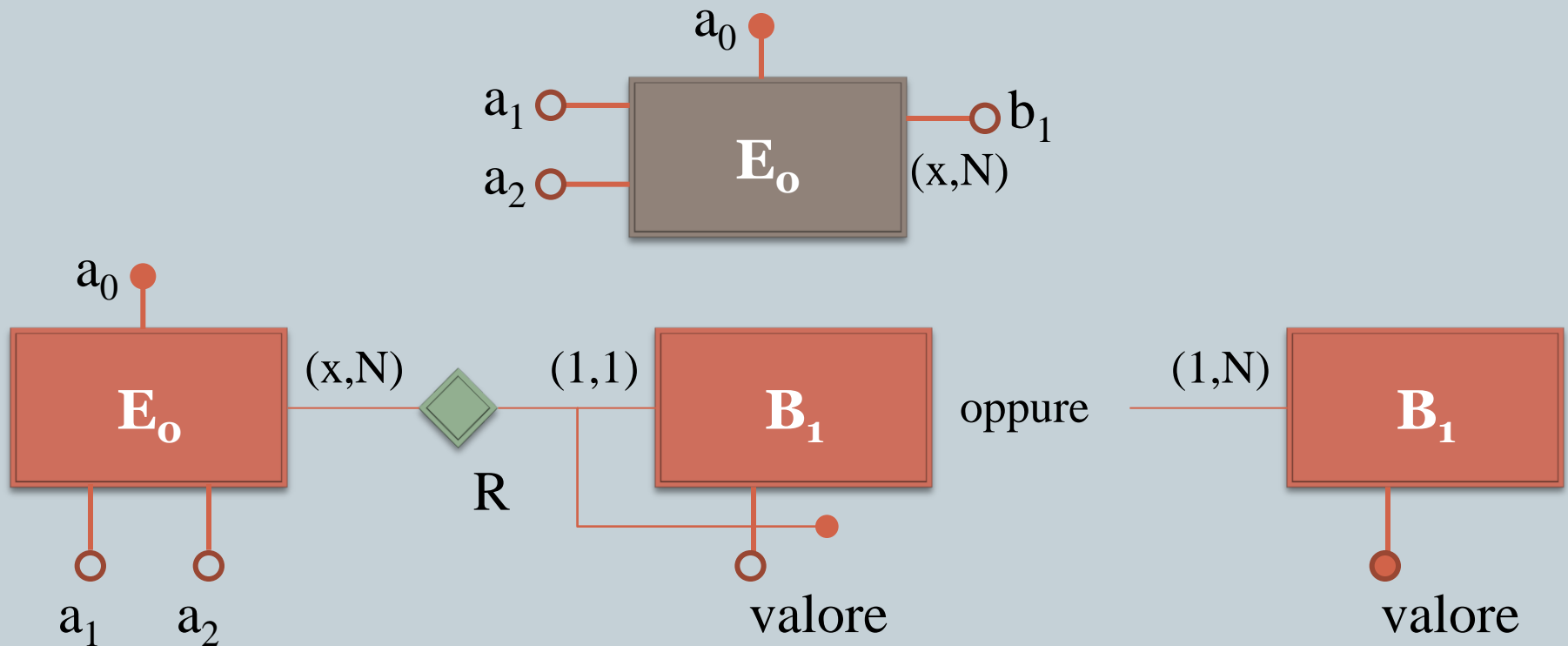
Si esegue solo nel caso in cui esistano operazioni frequenti che trattano solo un sottoinsieme degli attributi dell'entità da partizionare.



# Partizionamento di entità

20

## Eliminazione degli attributi multivalore



# Partizionamento di relazioni

21

Si realizza sostituendo una relazione  $R$  con due relazioni distinte  $R_1$  e  $R_2$ , dove le occorrenze di  $R$  si partizionano tra  $R_1$  e  $R_2$ .

E' conveniente quando esistono operazioni che si riferiscono ad un sottoinsieme delle occorrenze di  $R$  e altre operazioni che si riferiscono alle altre occorrenze di  $R$ .

# Scelta degli indentificatori principali

22

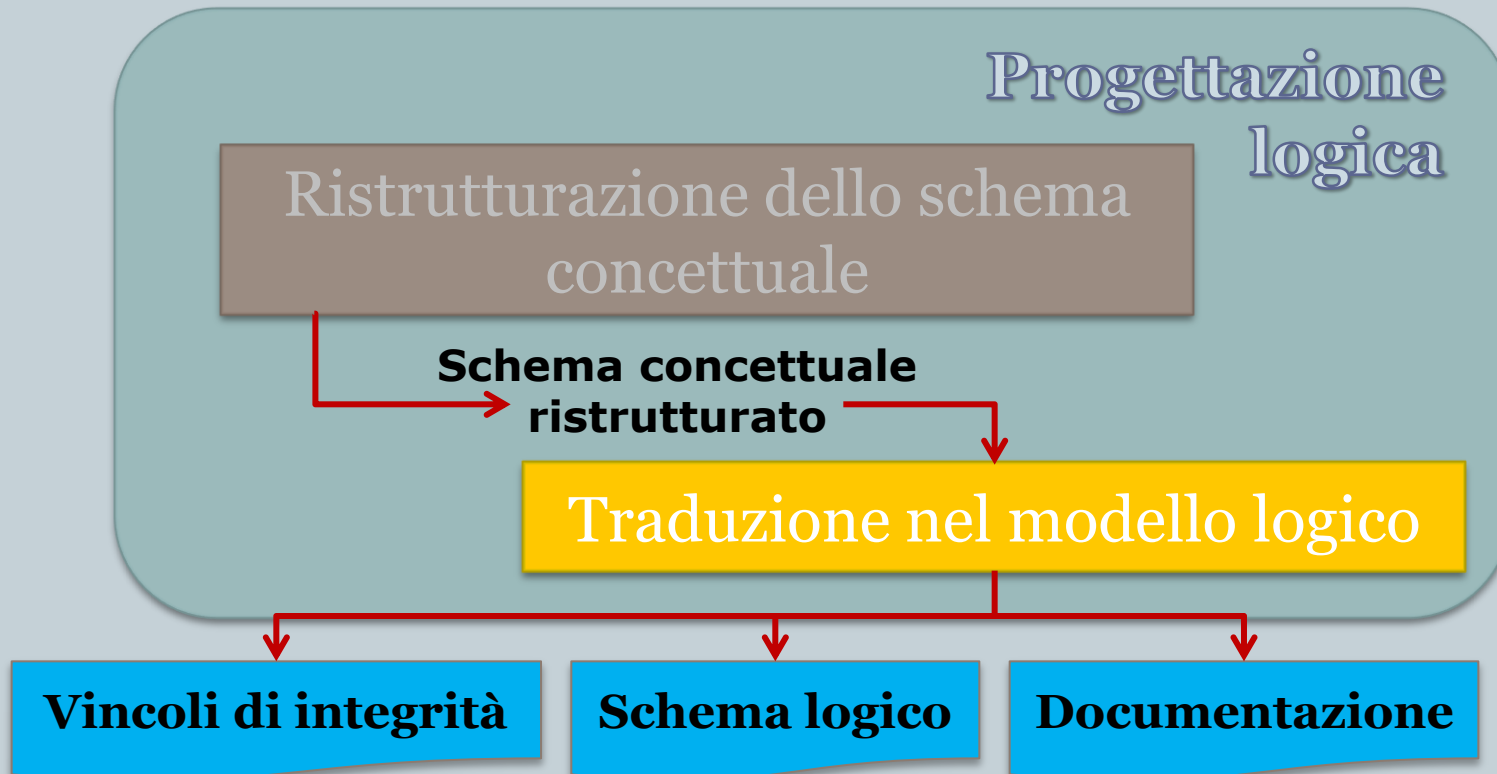
Va scelto un identificatore principale per ogni entità in quanto tale identificatore verrà utilizzato nel modello relazionale per identificare le tuple che rappresentano istanze di entità.

Criteri:

- Gli identificatori che includono attributi opzionali non possono essere chiavi primarie
- Meglio identificatori con pochi attributi
- Meglio identificatori utilizzati nelle operazioni

# Traduzione verso il modello relazionale

23



# Traduzione verso il modello relazionale

24

La traduzione è un processo automatico che si realizza applicando allo schema concettuale ristrutturato un insieme di regole di traduzione.

Ogni regola si applica ad un costrutto del modello concettuale ER e produce una o più strutture del modello relazionale.

In generale,

- si rappresenta un'istanza di entità con una tupla.
- si rappresenta un'istanza di relazione con un legame tra tuple o con una tupla esplicita

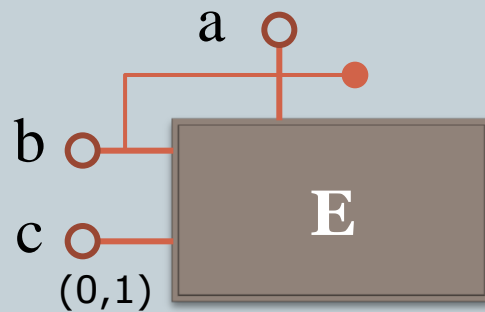
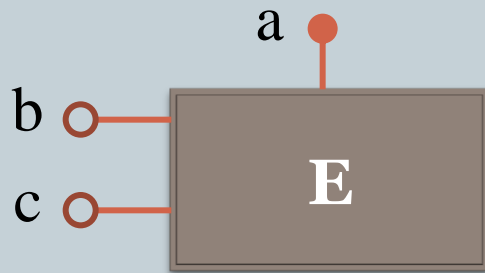


# Regole di traduzione

25

*modello ER*

R1 - ENTITÀ



*modello relazionale*

**E**(a, b, c)

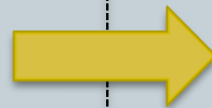
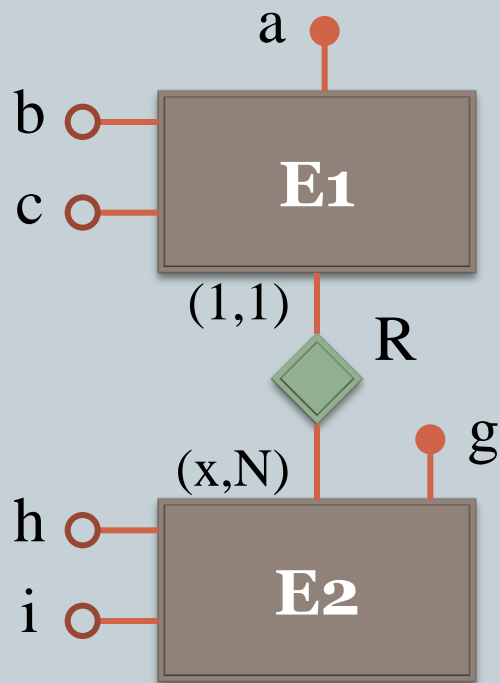
**E**(a, b, c\*)

# Regole di traduzione

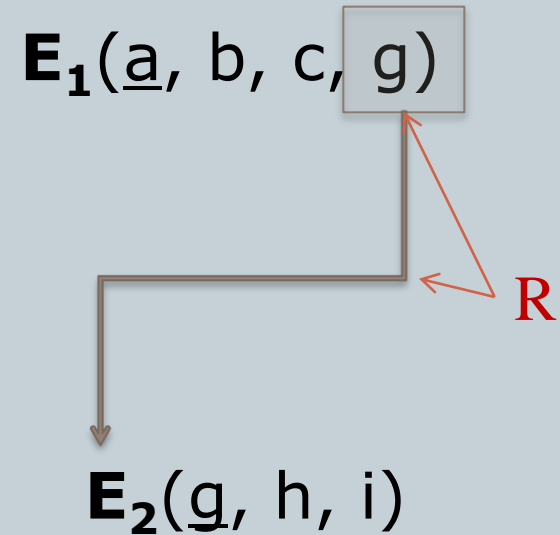
26

*modello ER*

R2 - RELAZIONE uno a molti



*modello relazionale*

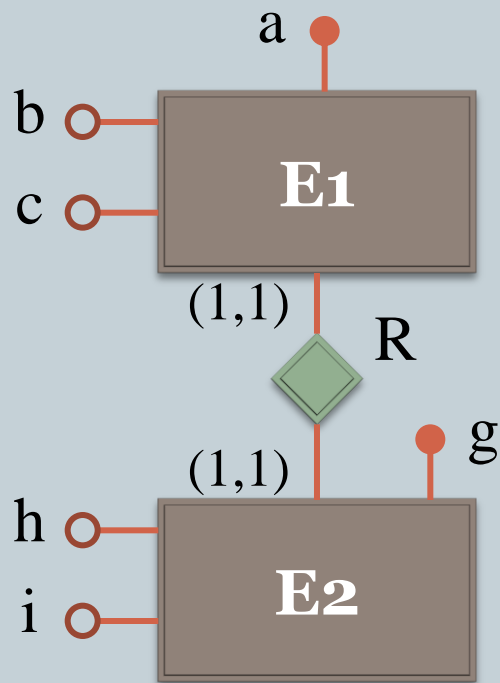


# Regole di traduzione

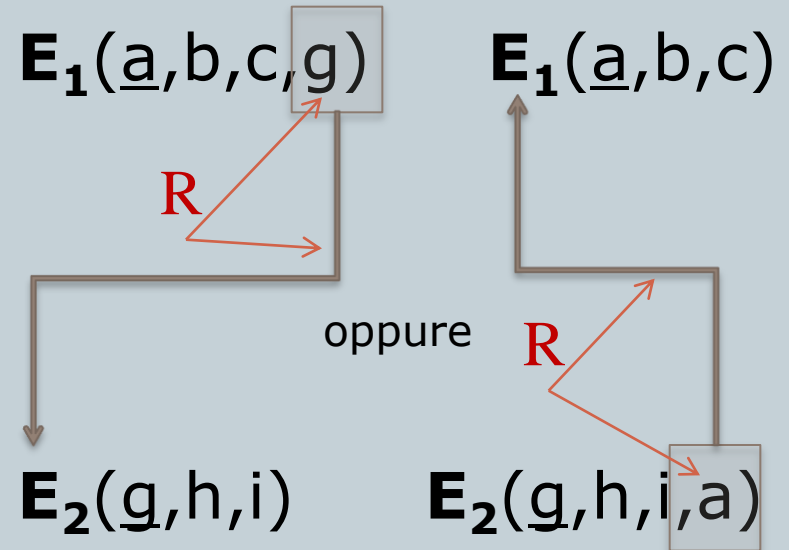
27

*modello ER*

R<sub>3</sub> - RELAZIONE uno a uno



*modello relazionale*

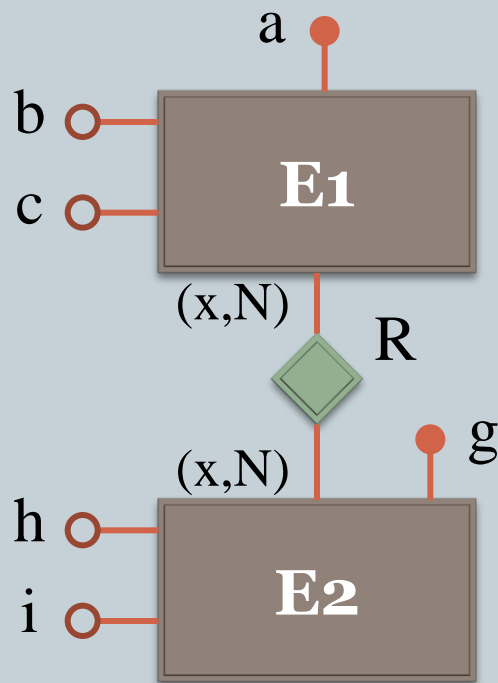


# Regole di traduzione

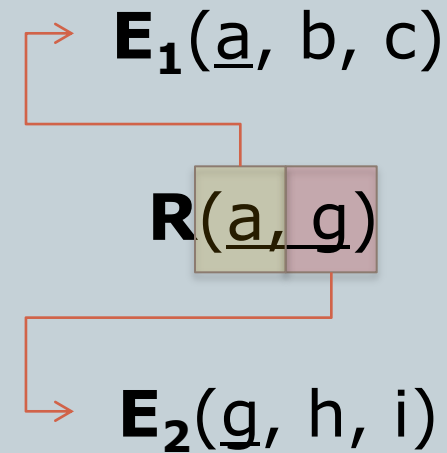
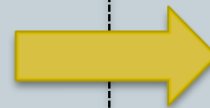
28

*modello ER*

R<sub>4</sub> - RELAZIONE molti a molti



*modello relazionale*

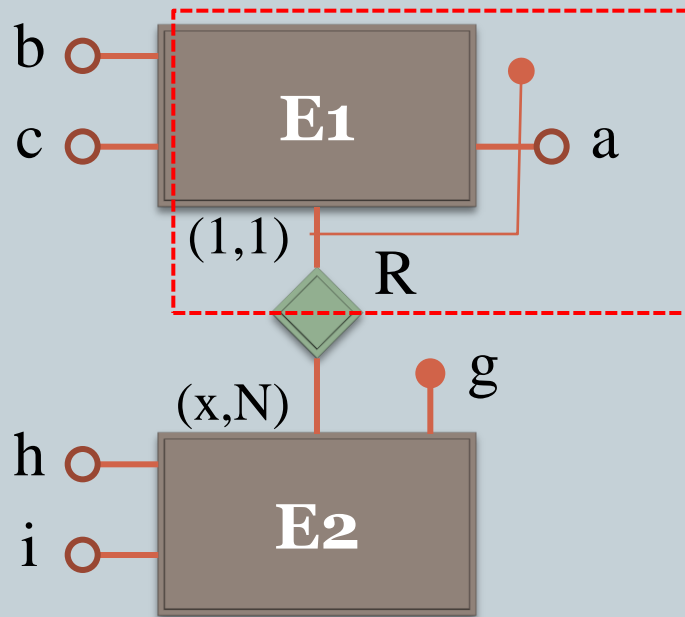


# Regole di traduzione

29

*modello ER*

R2.a - RELAZIONE uno a molti  
(identificatore esterno)



*modello relazionale*

$\mathbf{E}_1(a, \underline{g}, b, c)$

$\mathbf{E}_2(g, h, i)$

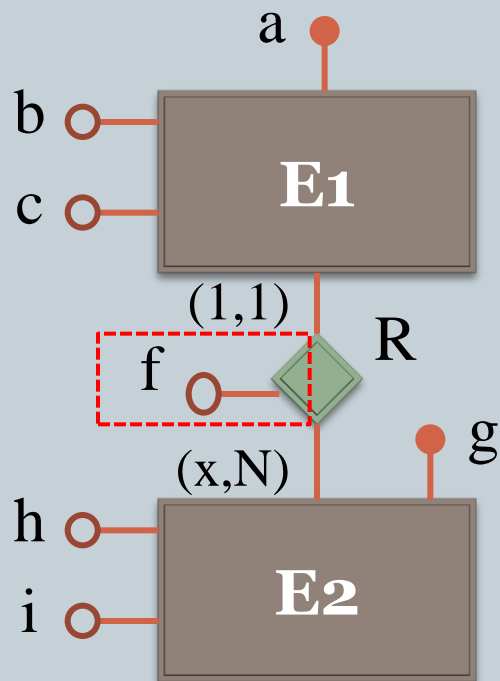
The relational model diagram shows the translation of the ER diagram. It consists of two tables:  $\mathbf{E}_1(a, \underline{g}, b, c)$  and  $\mathbf{E}_2(g, h, i)$ . The attribute  $\underline{g}$  in  $\mathbf{E}_1$  is underlined, indicating it is the primary key. A red arrow labeled R points from the primary key  $\underline{g}$  in  $\mathbf{E}_1$  to the attribute  $g$  in  $\mathbf{E}_2$ , representing the foreign key relationship.

# Regole di traduzione

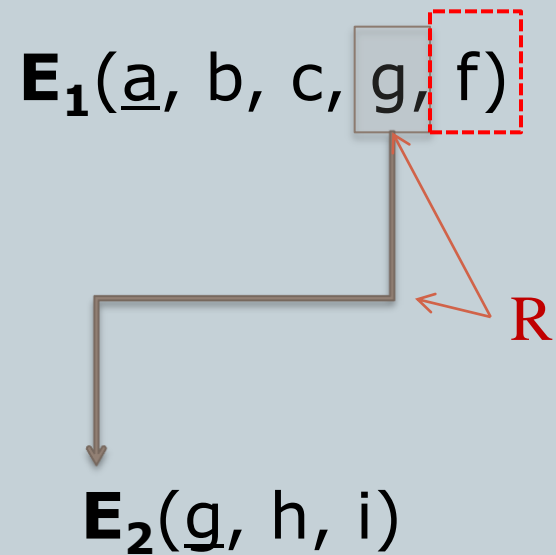
30

*modello ER*

R2.b - RELAZIONE uno a molti  
(con attributo sulla relazione)



*modello relazionale*

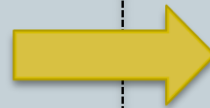
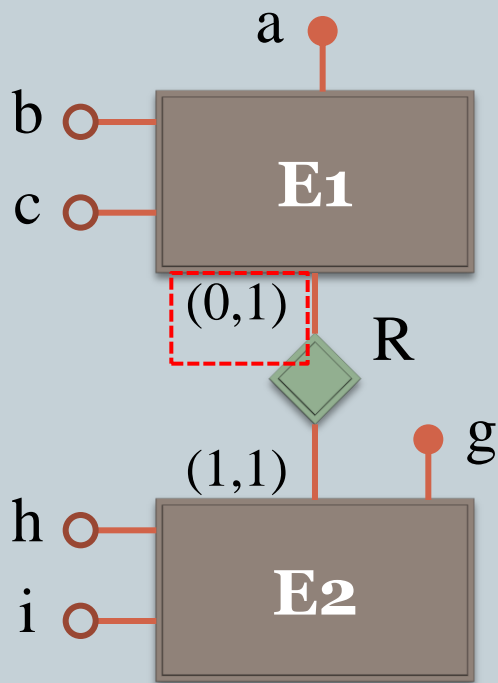


# Regole di traduzione

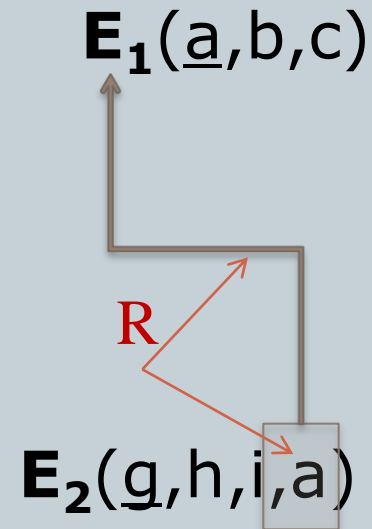
31

*modello ER*

R3.a - RELAZIONE uno a uno  
(con una cardinalità minima = 0)



*modello relazionale*

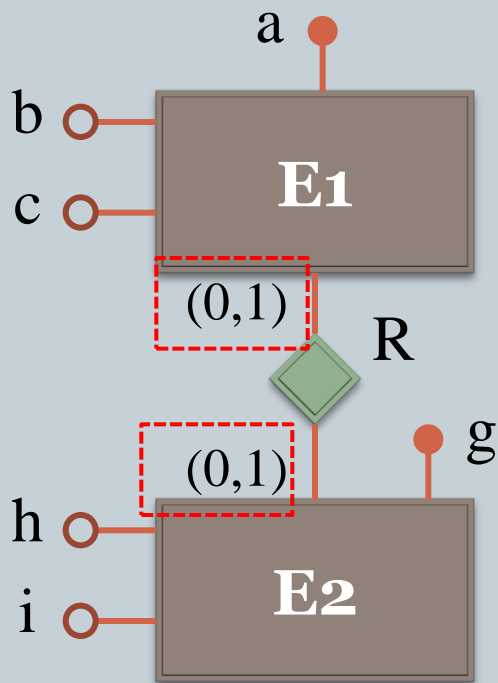


# Regole di traduzione

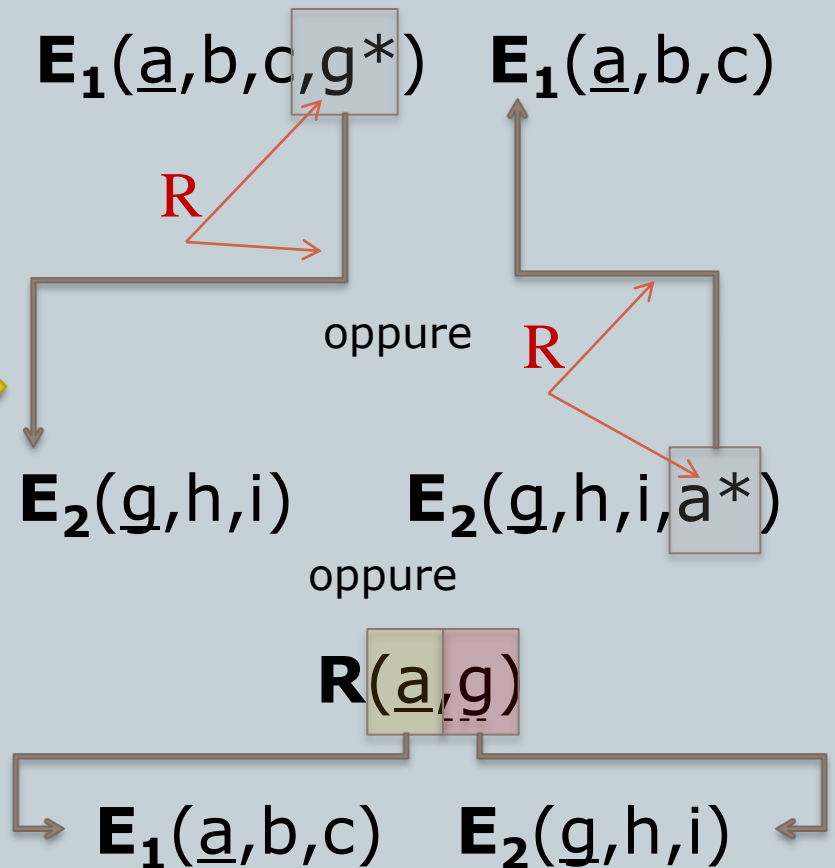
32

*modello ER*

R<sub>3.b</sub> - RELAZIONE uno a uno  
(con due cardinalità minima = 0)



*modello relazionale*



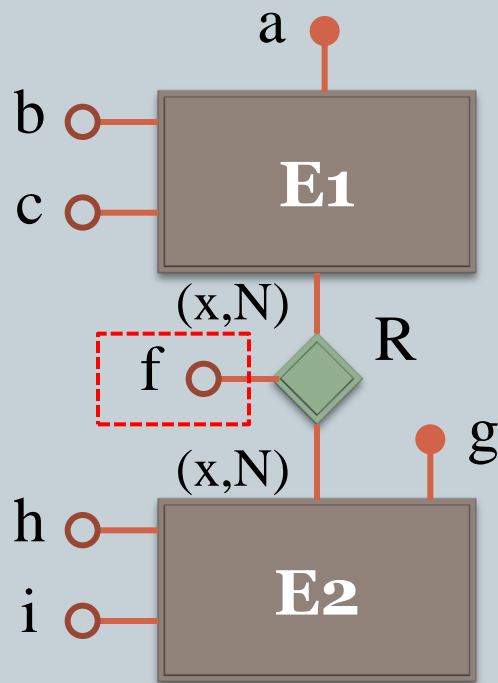


# Regole di traduzione

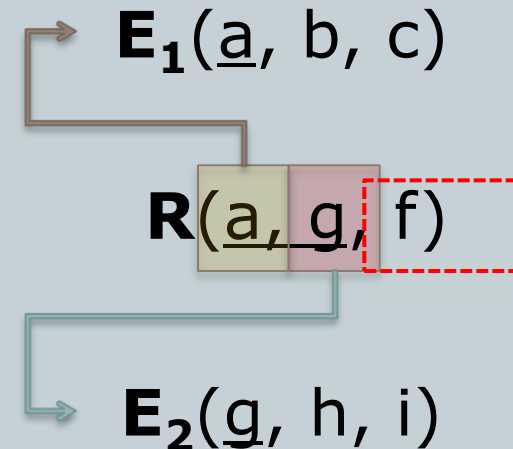
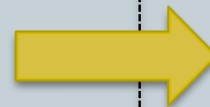
33

*modello ER*

R4.a - RELAZIONE molti a molti  
(con attributi)



*modello relazionale*

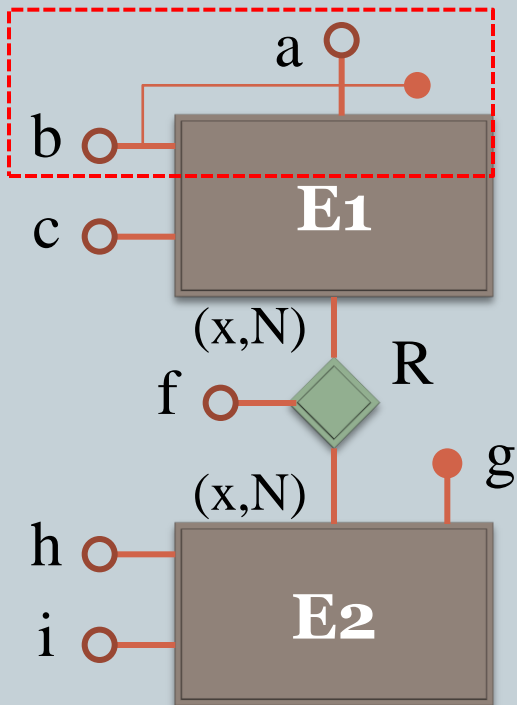


# Regole di traduzione

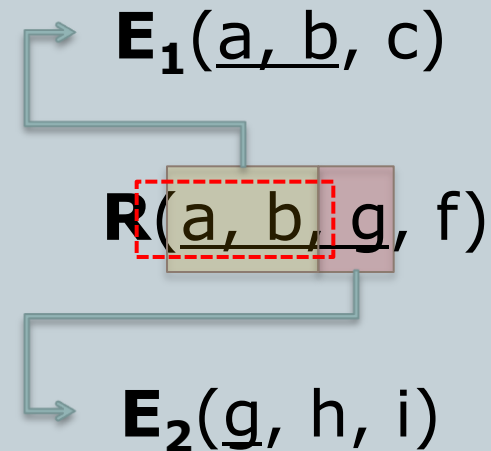
34

*modello ER*

R4.a' - RELAZIONE molti a molti  
(con identificatori con più attrib.)



*modello relazionale*

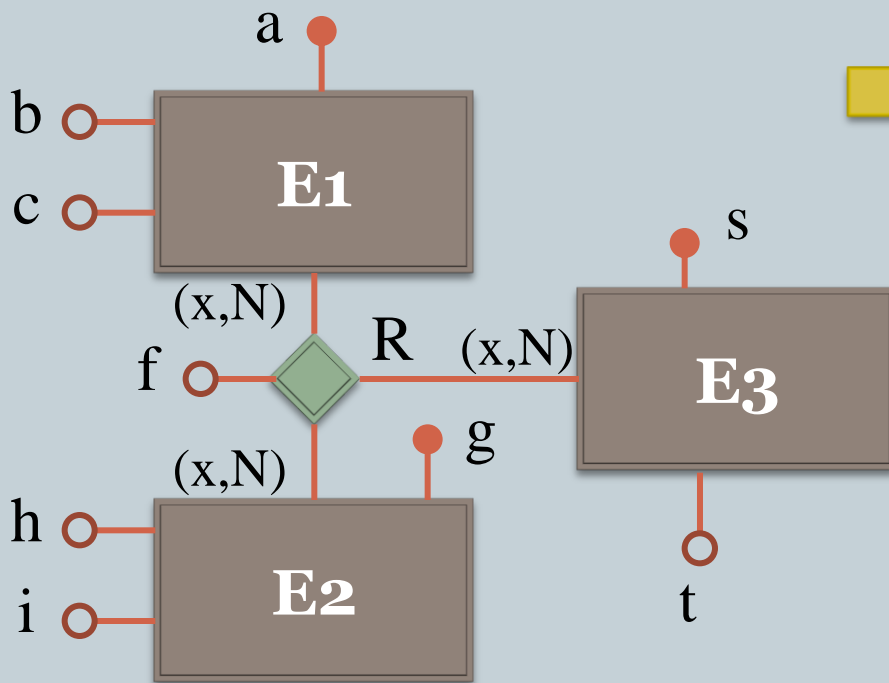


# Regole di traduzione

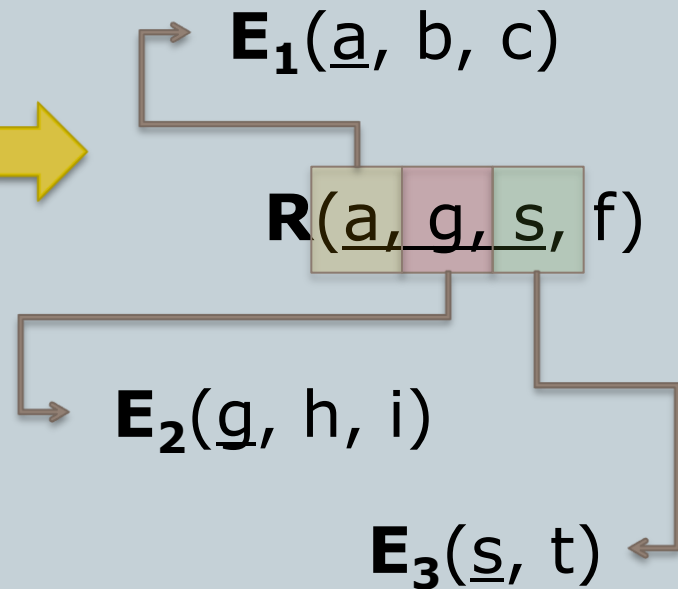
35

*modello ER*

R4.b - RELAZIONE molti a molti  
(relazione ternaria)



*modello relazionale*

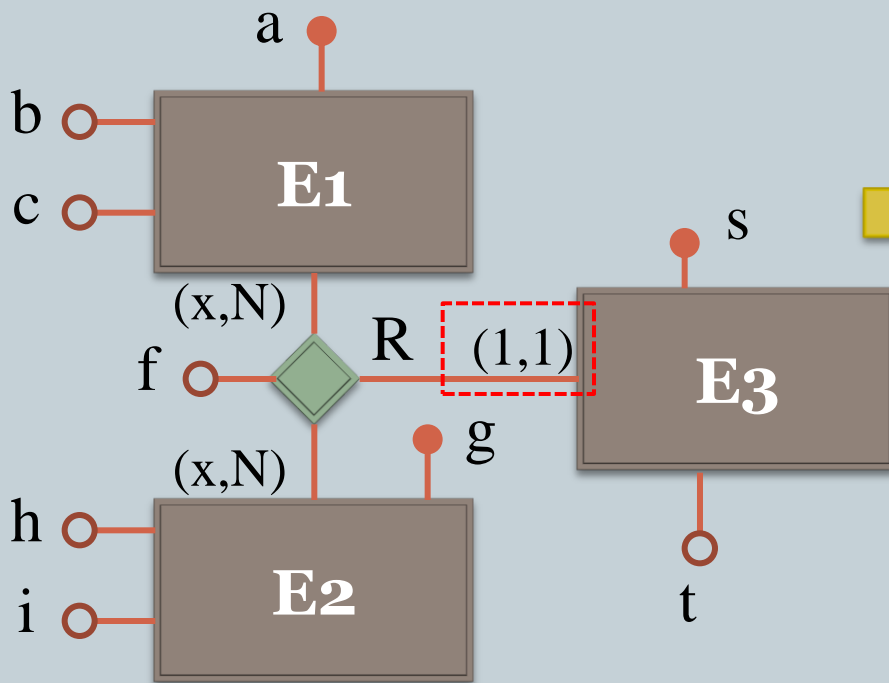


# Regole di traduzione

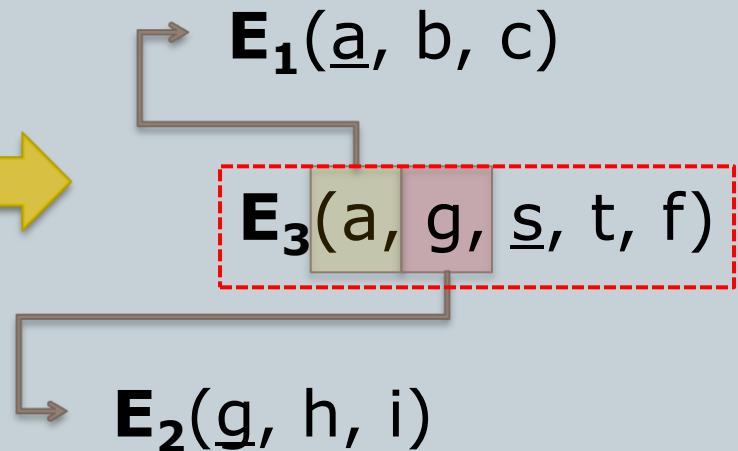
36

*modello ER*

R4.c - RELAZIONE molti a molti  
(relazione ternaria con card. (1,1))



*modello relazionale*



# Esercizio 2.7

37

## Pazienti

Cod	Cognome	Nome
A102	Necchi	Luca
B372	Rossini	Piero
B543	Missoni	Nadia
B444	Missoni	Luigi
S555	Rossetti	Gino

## Medici

Matr	Cognome	Nome	Reparto
203	Neri	Piero	A
574	Bisi	Mario	B
461	Bargio	Sergio	B
530	Belli	Nicola	C
405	Mizzi	Nicola	A
501	Monti	Mario	A

## Reparti

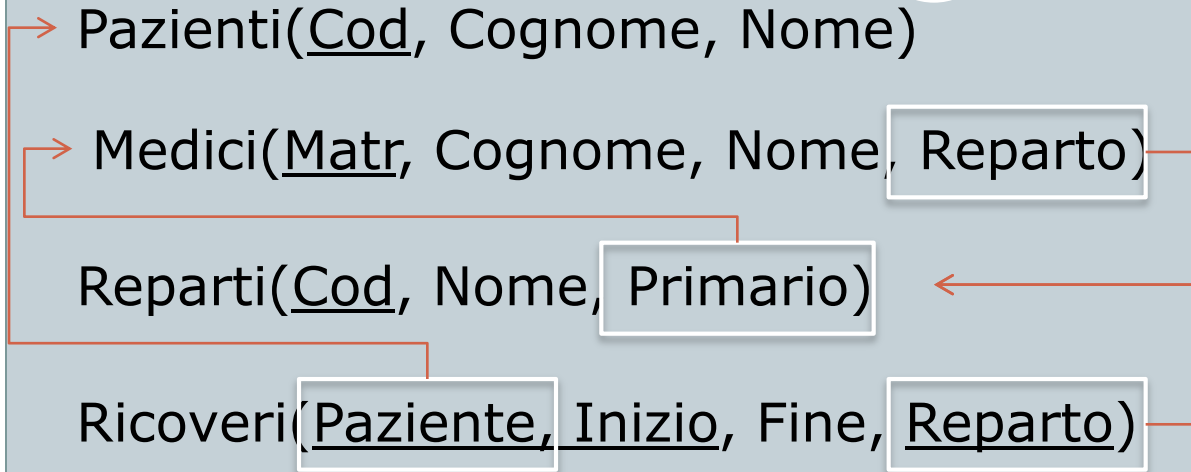
Cod	Nome	Primario
A	Chirurgia	203
B	Pediatria	574
C	Medicina	530

## Ricoveri

Paziente	Inizio	Fine	Reparto
A102	2/05/94	9/05/94	A
A102	2/12/94	2/01/95	A
S555	5/10/94	3/12/94	B
B444	1/12/94	2/01/95	B
S555	5/10/94	1/11/94	A

# Esercizio 2.7

38



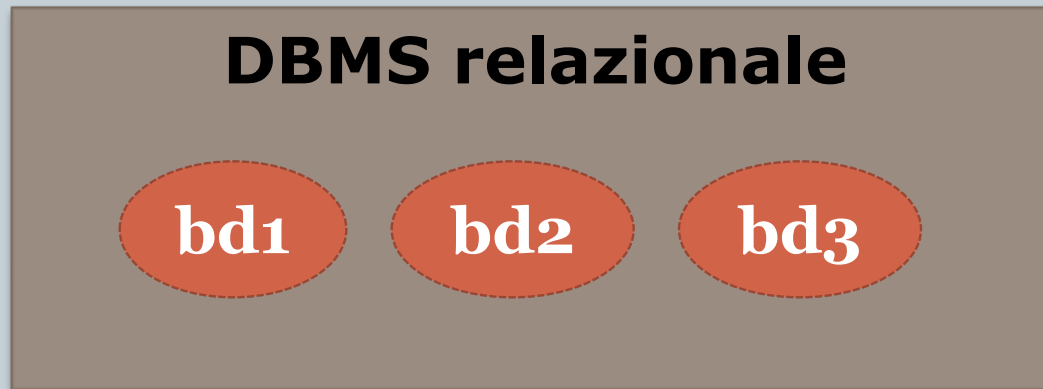
# Esercizi

39

- Esercizio: ristrutturare lo schema concettuale che descrive una compagnia aerea (ER.3) e tradurre nel corrispondente schema relazionale
- Esercizio: tradurre nel corrispondente schema relazionale lo schema concettuale che descrive una banca (ER.2)

# Operazioni su una base di dati relazionale

40



Data Definition Language (DDL):  
linguaggio per la definizione dei dati

Data Manipulation Language (DML):  
linguaggio per la manipolazione dei dati

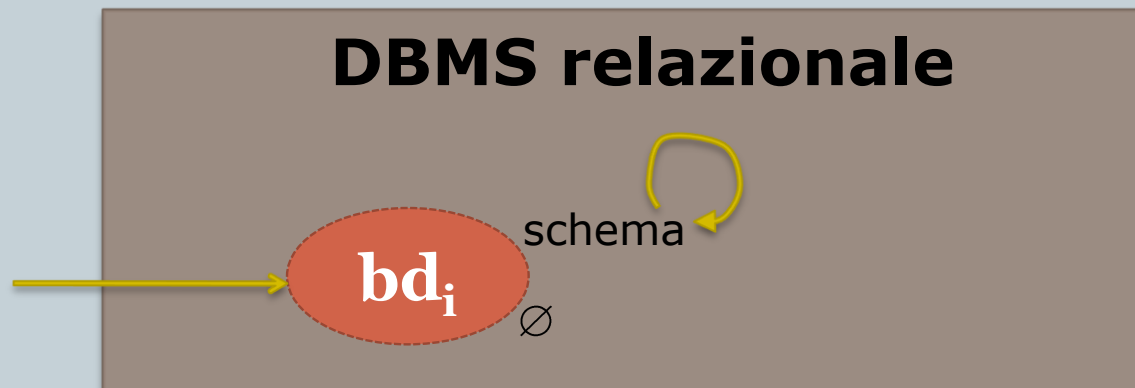


# Operazioni su una base di dati relazionale

41

DDL: consente di

- Creare lo schema della base di dati
- Modificare lo schema della base di dati

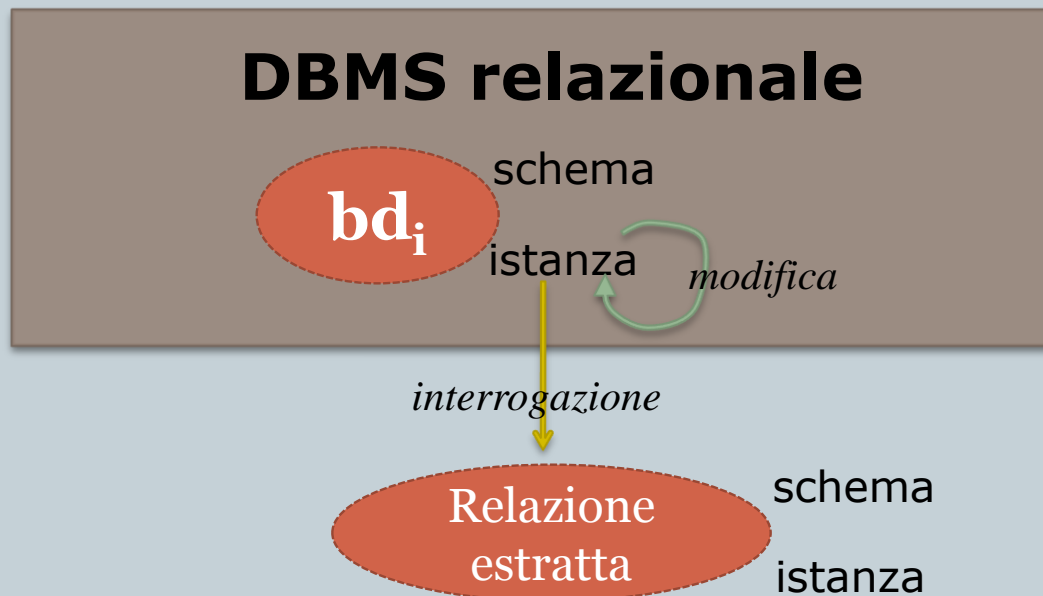


# Operazioni su una base di dati relazionale

42

DML: consente di

- Inserire/cancellare tuple nelle relazioni
- Aggiornare i valori nelle tuple delle relazioni
- Interrogare le relazioni per estrarre informazione



# Linguaggi di interrogazione

43

## Classificazione

- Linguaggi di tipo procedurale: specificano il procedimento per ottenere il risultato.
- Linguaggi di tipo dichiarativo: specificano le proprietà del risultato.

Linguaggi procedurali: ALGEBRA RELAZIONALE

Linguaggi dichiarativi: CALCOLO RELAZIONALE e  
STRUCTURED QUERY LANGUAGE (SQL)