

# Progettazione logica - da schemi ER a schemi relazionali

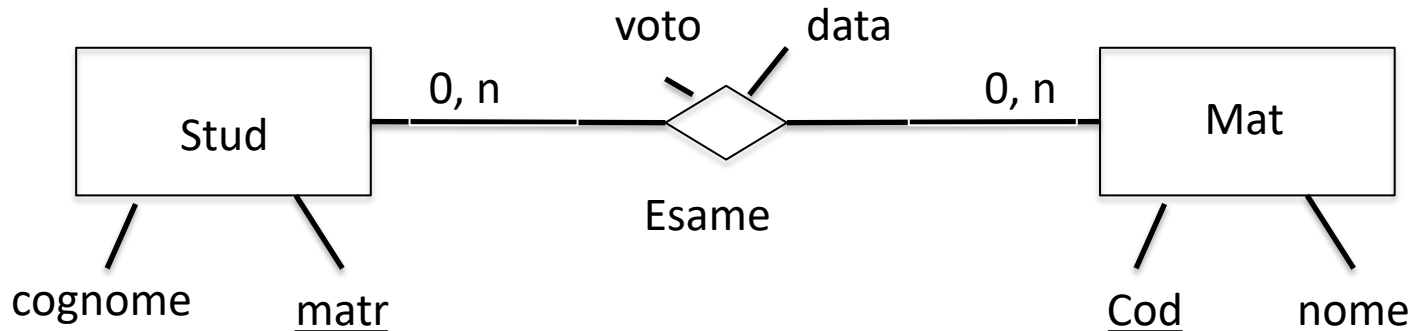
P. Rullo

[rullo@unical.it](mailto:rullo@unical.it)

# Progettazione logica

- Lo schema concettuale ER viene trasformato in uno schema relazionale equivalente
- Si tratta di rappresentare attraverso l'unico costrutto del modello dei dati relazionale i vari costrutti del modello ER
- La trasformazione si basa su semplici regole di "sintattiche"

# Associazioni n:m



- Schema relazionale equivalente
  - **Stud(matr, cognome)**
  - **Materia(cod, nome)**
  - **Esame(matr\*, mat\*, data, voto)**
- Una relazione per ogni entità, più una relazione per l'associazione
- Gli attributi *matr* e *mat* di Esame sono chiavi secondarie che si riferiscono alle chiavi primarie *matr* (Stud) e *Cod* (Materia)
- La coppia di attributi (*matr*, *mat*) di Esame è chiave primaria – quindi non possono esistere più registrazioni dello stesso esame

# Associazioni n:m

- Schema relazionale
  - Stud(matr, cognome)
  - Materia(cod, nome)
  - Esame(matr\*, mat\*, data, voto)

Stud

<b>matr</b>	<b>cognome</b>
252	neri
333	rossi
888	bianchi

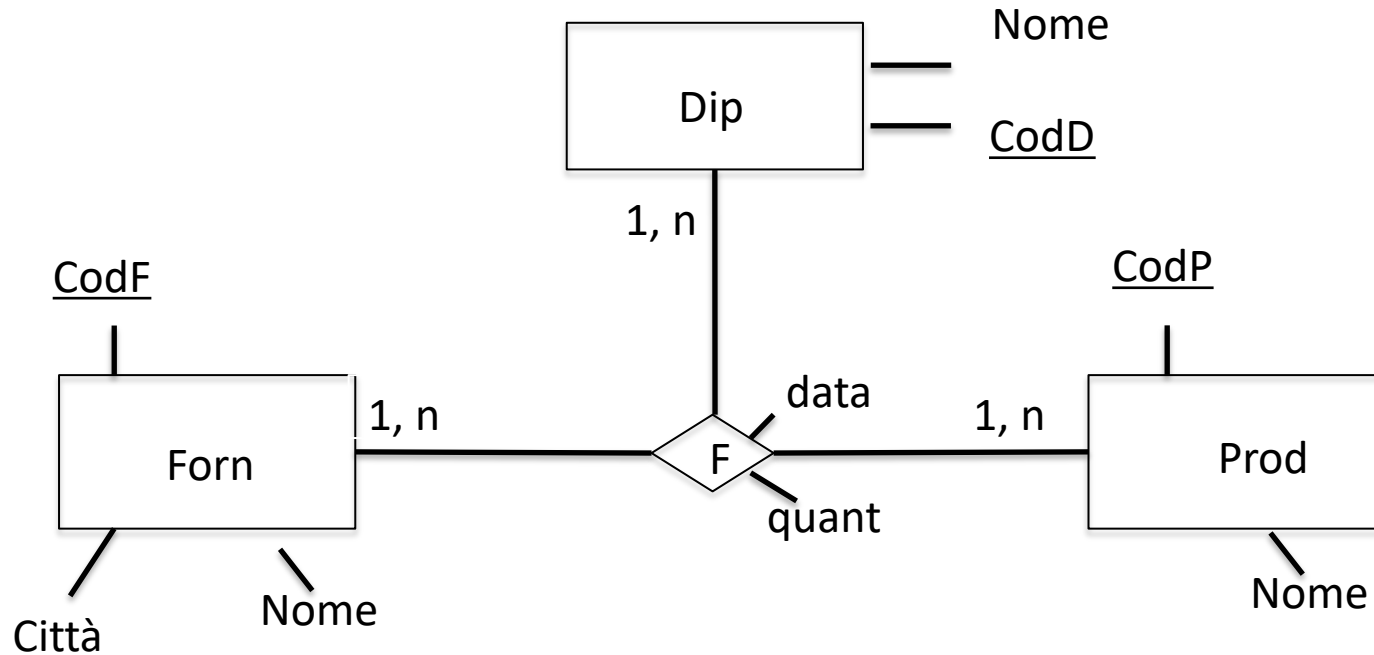
Materia

<b>cod</b>	<b>nome</b>
2b	Basi dati
3b	Arch
4b	PO

Esame

<b>matr</b>	<b>mat</b>	<b>data</b>	<b>voto</b>
252	2b	3/3/18	30
252	3b	2/2/19	18
333	4b	1/1/19	24

# Associazioni n:m

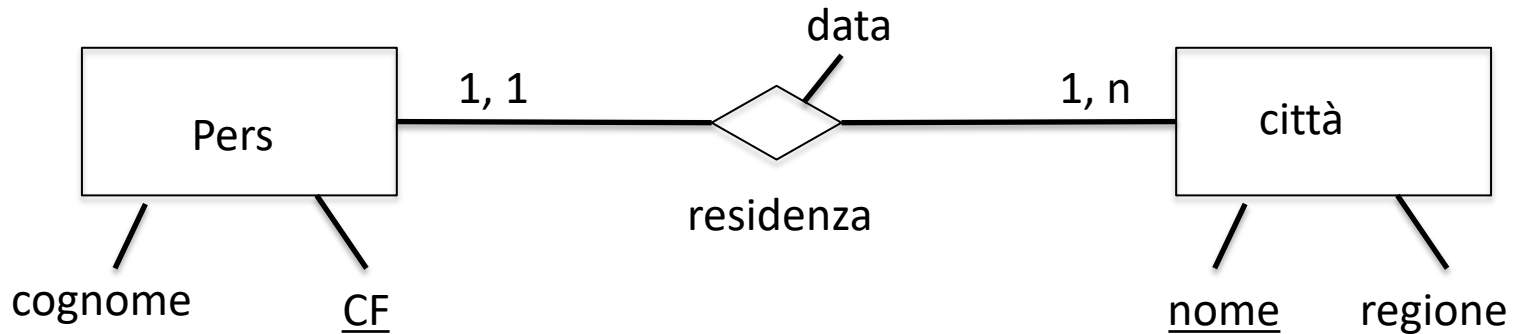


- **Forn(codF, città, nome)**
- **Dip(CodD, nome)**
- **Prod(CodP, nome)**
- **Fornitura(codF\*, codP\*, codD\*, quant, data)**

# Associazioni n:m

- Lo schema di una relazione R che rappresenta una associazione n-aria multi-a-molti del modello ER è composto
  - dalle n chiavi primarie delle entità connesse; ognuna di queste è una chiave secondaria, e tutte assieme formano la chiave primaria di R
  - dagli attributi della associazione

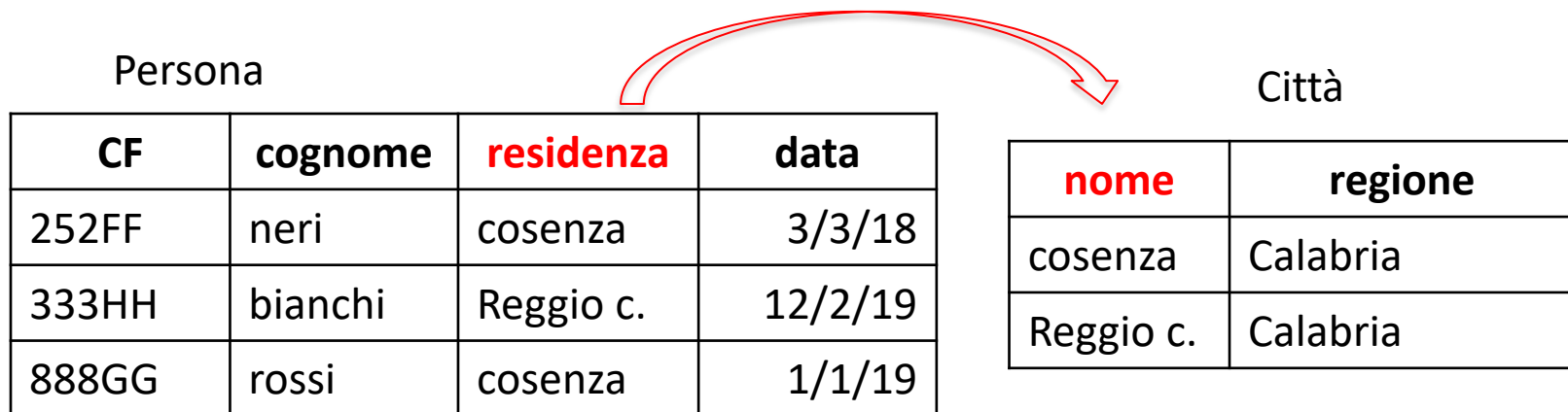
# Associazioni 1:n



- Schema relazionale equivalente
  - **Pers(CF, cognome, residenza\*, data)**
  - **Città(nome, regione)**
- L'associazione *residenza* si rappresenta con l'attributo *residenza* di Pers che è chiave secondaria che si riferisce alla chiave primaria *nome* di Città - esso consente quindi di associare ad ogni istanza di persona una istanza di città
- L'attributo *data* della relazione *residenza* viene inserito nella relazione Pers in cui è presente la chiave secondaria *residenza*

# Associazioni 1:n

- Schema relazionale equivalente
  - **Pers(CF, cognome, residenza\*, data)**
  - **Città(nome, regione)**





# Associazioni 1:n

- Schema relazionale equivalente
  - **Pers(CF, cognome, residenza\*, data)**
  - **Città(nome, regione)**
- con vincolo di integrità
  - valore NULL NON ammesso per residenza – ogni persona *deve* avere una residenza (associazione *obbligatoria*)

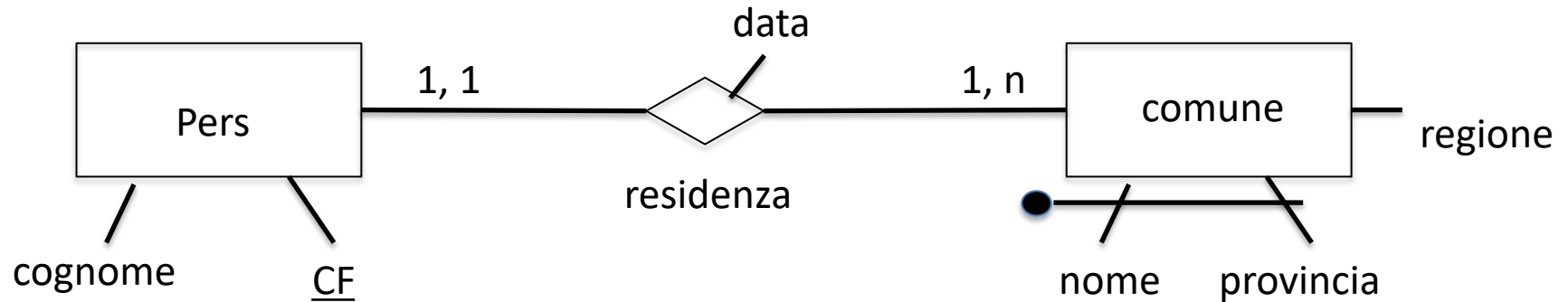
Persona

CF	cognome	residenza	data
252FF	neri	NULL	3/3/18
333HH	bianchi	Reggio c.	12/2/19
888GG	rossi	cosenza	1/1/19

Città

nome	regione
cosenza	Calabria
Reggio c.	Calabria

# Associazioni 1:n

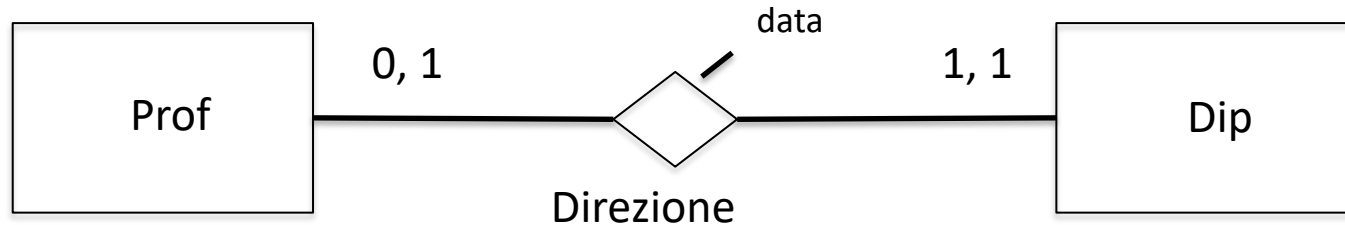


- Schema relazionale
  - **Pers(CF, cognome, <nome, provincia>\*, data)**
  - **Comune(nome, provincia, regione)**
- La chiave secondaria in Pers è <nome, provincia> - essa consente di associare ad ogni istanza di persona una istanza di comune

# Associazioni 1:n - riassunto

- Una associazione 1:n tra due entità A e B, dove A sta dalla parte 1 della associazione, si rappresenta introducendo nella relazione che rappresenta A
  - una chiave secondaria S definita sulla chiave primaria della relazione che rappresenta B
  - gli eventuali attributi della associazione
  - se l'associazione è *obbligatoria*, cioè, il vincolo di cardinalità minima dal lato di A è pari a 1, un vincolo che vieta valori nulli sulla chiave secondaria S

# Associazioni 1:1



## SCHEMA1

- **Prof**(codP, nome, età)
- **Dip**(codD, nome, ha\_dir\*, data)

## SCHEMA2

- **Prof**(codP, nome, età, è\_dir\*, data)
- **Dip**(codD, nome)

- Come nel caso generale, l'associazione si rappresenta tramite chiave secondaria. Essendo l'associazione simmetrica (1:1) si può scegliere dove piazzarla – in **Prof** o in **Dip**
- I due schemi così ottenuti sono equivalenti

# Associazioni 1:1

## SCHEMA1

- Prof(codP, nome, età)
- Dip(codD, nome, ha\_dir\*, data)

CodP	nome	età
252	neri	33
333	bianchi	44
999	rossi	55

- Secondo questo schema, ad ogni Dip è associato un *unico* direttore attraverso la chiave secondaria *ha\_dir*

CodD	nome	ha_dir	data
d1	Demacs	252	1/1/19
d2	Dimes	999	1/2/18

# Associazioni 1:1

## SCHEMA1

- Prof(codP, nome, età)
- Dip(codD, nome, ha\_dir\*, data)

CodP	nome	età
252	neri	33
333	bianchi	44
999	rossi	55

CodD	nome	ha_dir	data
d1	Demacs	252	1/1/19
d2	Dimes	999	1/2/18
d3	Dibest	<del>252</del>	3/3/20

- Secondo questo schema, ad ogni Dip è associato un *unico* direttore attraverso la chiave secondaria *ha\_dir*
- Per garantire che l'associazione sia 1:1 e, quindi, che un Prof NON sia direttore di più di un Dip, si deve introdurre un vincolo di **unicità** sulla chiave secondaria *ha\_dir*

# Associazioni 1:1

## SCHEMA1

- Prof(codP, nome, età)
- Dip(codD, nome, ha\_dir\*, data)

CodP	nome	età
252	neri	33
333	bianchi	44
999	rossi	55

- Per garantire che un Dip abbia sempre un direttore (associazione obbligatoria) è necessario NON ammettere **valore NULL** sulla chiave secondaria *ha\_dir*

CodD	nome	ha_dir	data
d1	Demacs	252	1/1/19
d2	Dimes	999	1/2/18
d3	Dibest	<del>NULL</del>	2/2/20

# Associazioni 1:1

## SCHEMA2

- Prof(codP, nome, età, è\_dir\*, data)
- Dip(codD, nome)

CodP	nome	età	è_dir	data
252	neri	33	d1	1/1/19
333	bianchi	44	NULL	NULL
999	rossi	55	d2	1/2/18

CodD	nome
d1	Demacs
d2	Dimes

- Secondo questo schema, ad ogni Prof è associato un *unico* Dip, attraverso la chiave secondaria *è\_dir*
- Tale chiave può assumere valore NULL in quanto non ogni Prof è direttore di qualche Dip (associazione opzionale lato Prof)



# Associazioni 1:1

## SCHEMA2

- Prof(codP, nome, età, è\_dir\*, data)
  - Dip(codD, nome)
- Per garantire che l'associazione sia 1:1 e, quindi, che un Dip NON abbia più di un direttore, si deve introdurre un vincolo di **unicità** sulla chiave secondaria *è\_dir*

CodP	nome	età	è_dir	data
252	neri	33	d1	1/1/19
333	bianchi	44	NULL	NULL
999	rossi	55	<del>d1</del>	1/2/18

CodD	nome
d1	Demacs
d2	Dimes

# Associazioni 1:1

## SCHEMA1

- Prof(codP, nome, età)
- Dip(codD, nome, ha\_dir\*, data)

## SCHEMA2

- Prof(codP, nome, età, è\_dir\*, data)
- Dip(codD, nome)

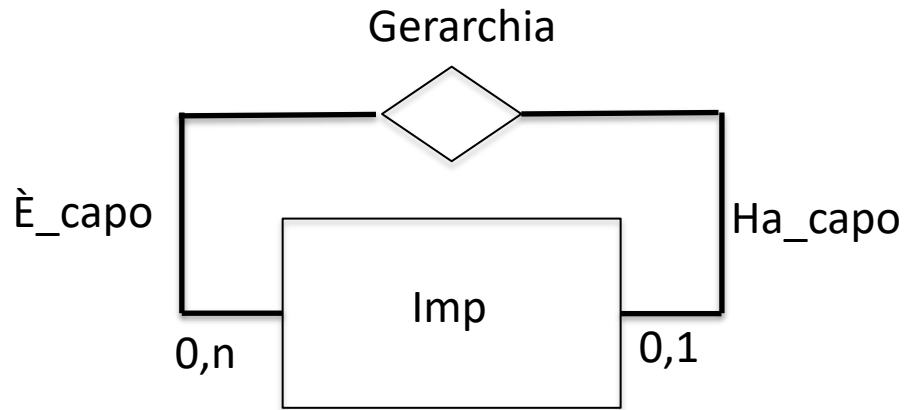
I due schemi a confronto:

- SCHEMA1 NON ammette valori nulli
- SCHEMA2 ammette valori nulli
- Ancorchè i due schemi siano equivalenti, SCHEMA1 è preferibile per via della mancanza di valori nulli

# Associazioni 1:1

- Si tratta di una associazione simmetrica
- Una associazione 1:1 tra due entità A e B si rappresenta introducendo nella relazione che rappresenta A (oppure B)
  - una chiave secondaria S definita sulla chiave primaria della relazione che rappresenta B (oppure A)
  - gli eventuali attributi della associazione
  - un vincolo di unicità sui valori di S
  - eventuale vincolo che vieta valori nulli su S, se l'associazione è *obbligatoria*

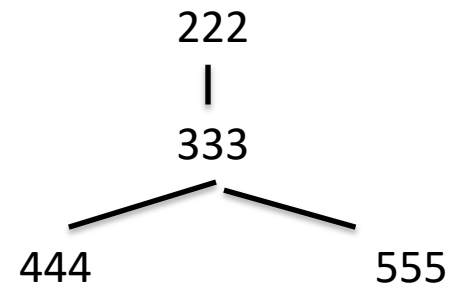
# Associazioni ricorsive



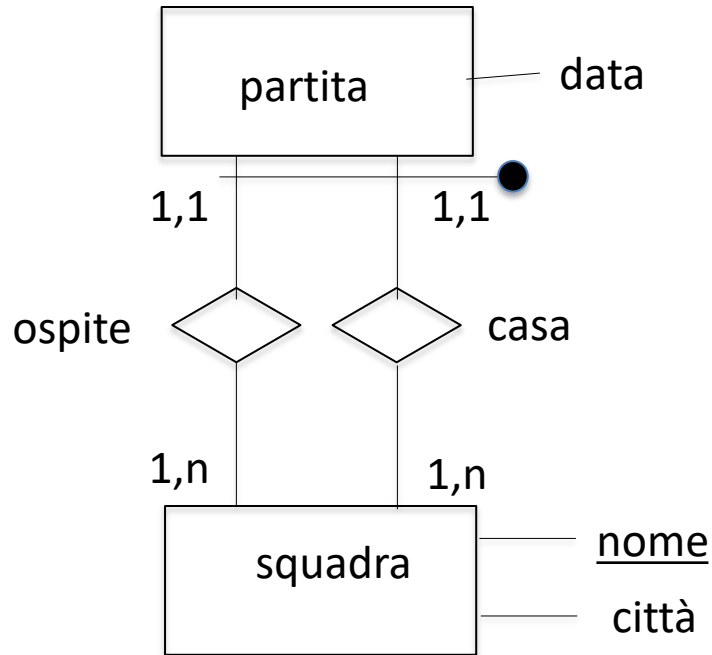
- NOTA: si tratta di una relazione 1:n
- Nella relazione Imp, l'attributo *ha\_capo* è una chiave secondaria che ammette come valori le matricole degli impiegati
- Quindi, una chiave secondaria definita sulla chiave primaria della stessa relazione di appartenenza

- **Imp(matr, nome, ha\_capo\*)**

matr	nome	ha_capo
222	maria	NULL
333	gianni	222
444	aldo	333
555	clara	333



# Chiavi Esterne (Composte)



- **Squadra(nome, città)**
- **Partita(casa\*, ospite\*, data)**

- La chiave primaria di Partita è la coppia di nomi delle due squadre
- *casa* e *ospite* sono anche chiavi secondarie che rappresentano le due associazioni 1:n

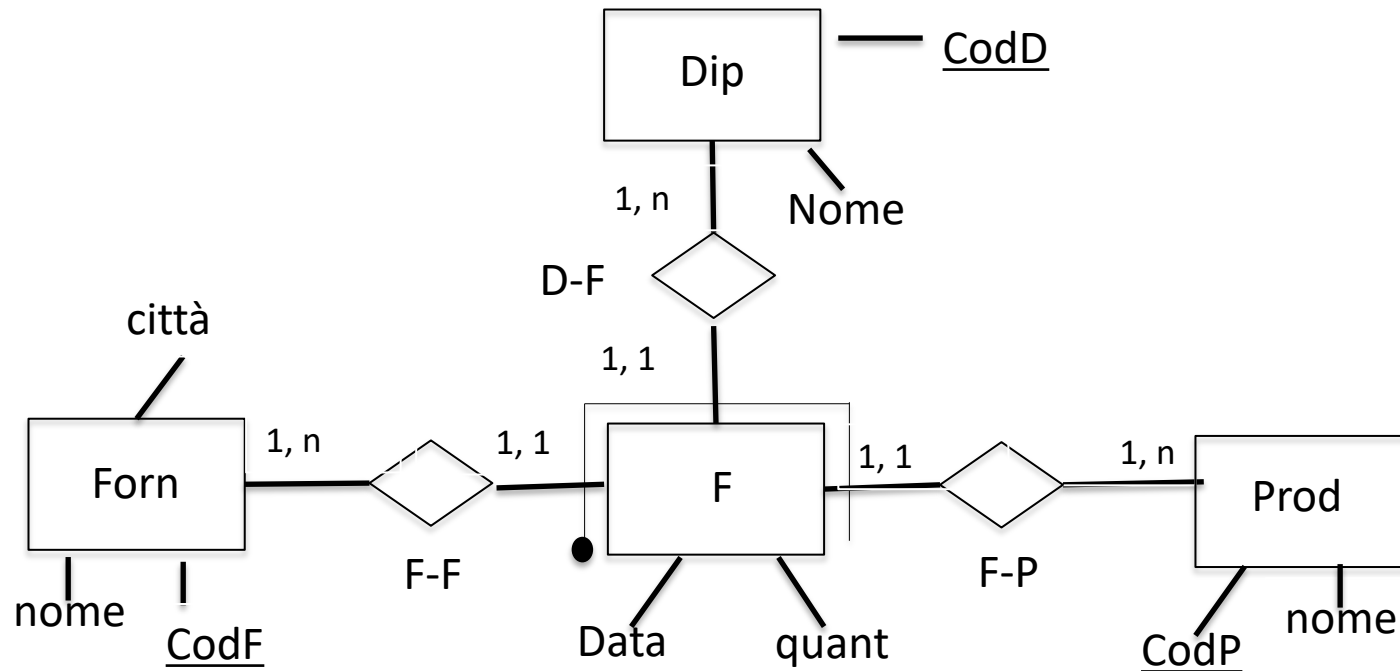
Squadra

Nome	Città
Juve	Torino
Milan	Milano
Inter	Milano

Partita

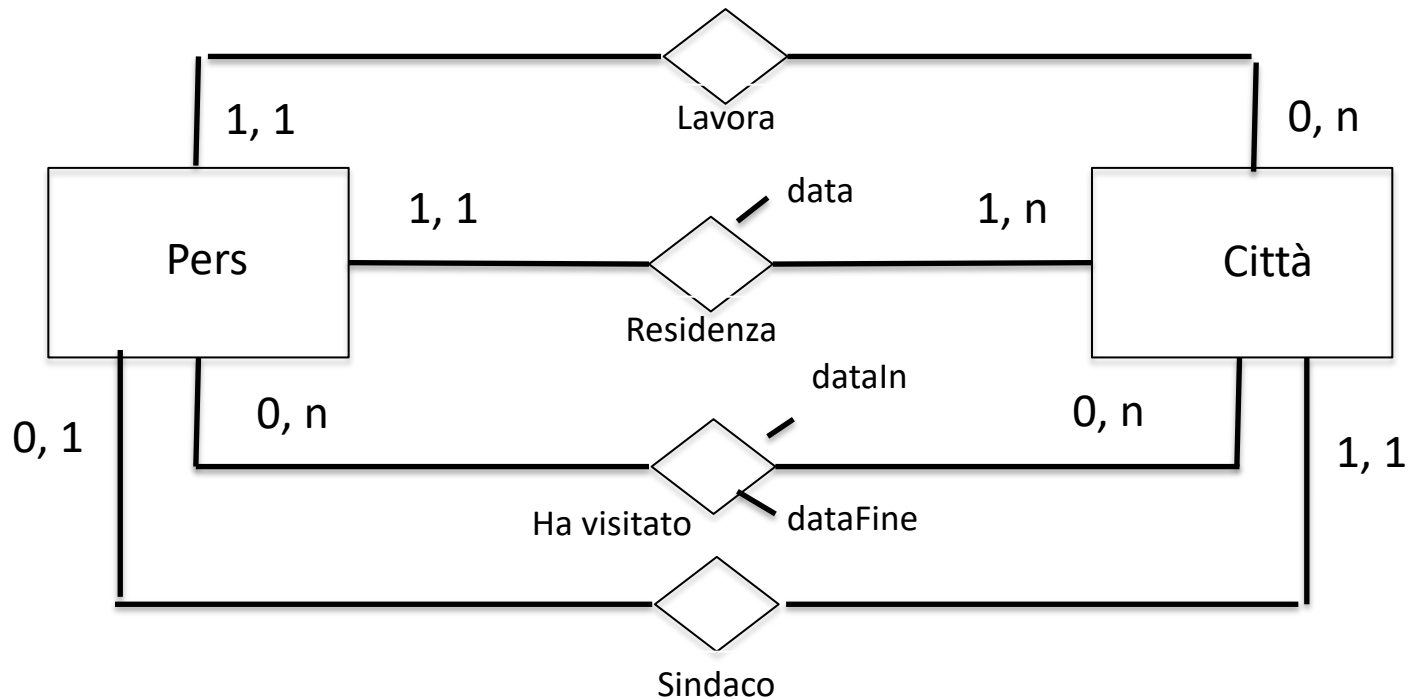
Casa	Ospite	data
Juve	Inter	3/3/19
Milan	Juve	2/2/20
Milan	Inter	1/1/20

# Chiavi Esterne (Composte)



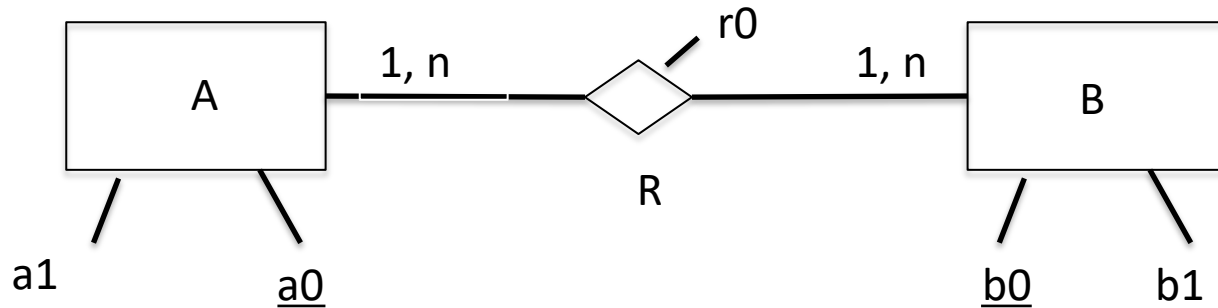
- **Forn**(codF, città, nome)
- **Dip**(CodD, nome)
- **Prod**(CodP, nome)
- **Fornitura**(codF\*, codP\*, codD\*, quant, data)

# Esempio di traduzione

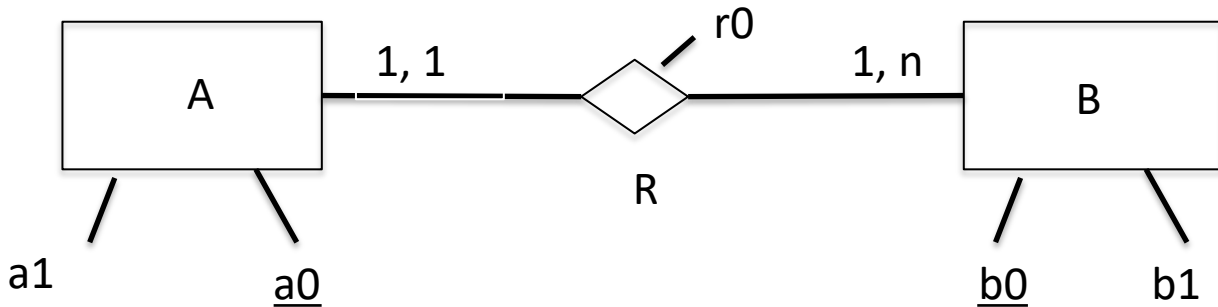


- **Pers**(CF, nome, resid\*, data, sedeLavoro\*)
- **Città**(nome, regione, sindaco\*)
- **HaVisitato**(persona\*, città\*, dataIn, dataFine)

# Quadro riassuntivo



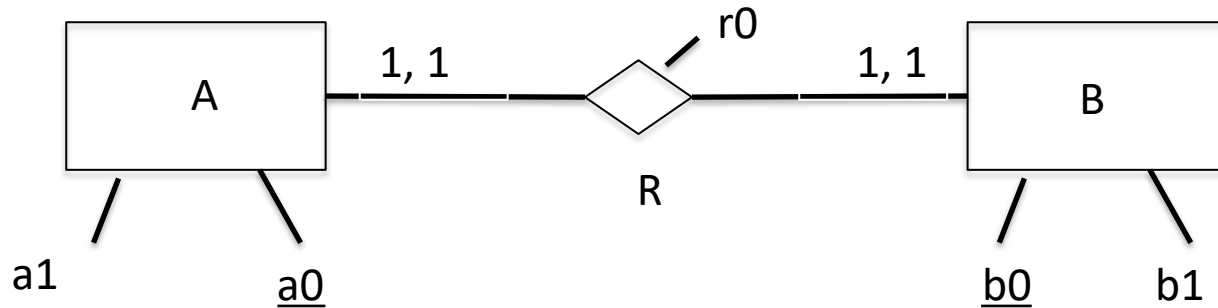
- $A(\underline{a_0}, a_1), B(\underline{b_0}, b_1), R(\underline{a_0^*}, \underline{b_0^*}, r_0)$



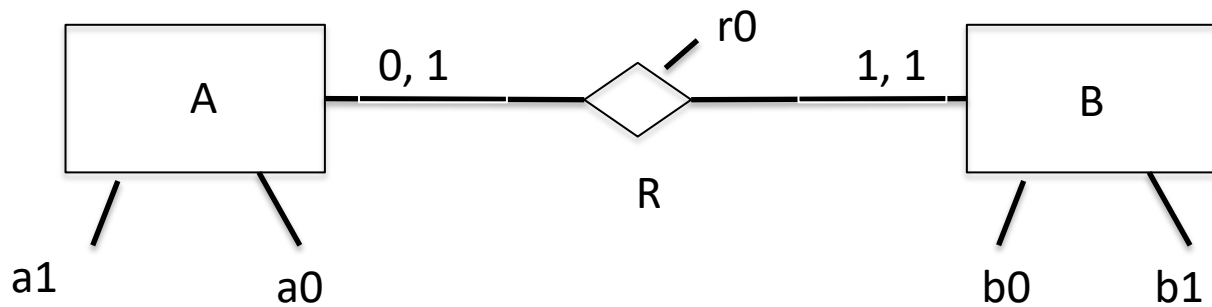
- $A(\underline{a_0}, a_1, \underline{b_0^*}, r_0), B(\underline{b_0}, b_1)$



# Quadro riassuntivo

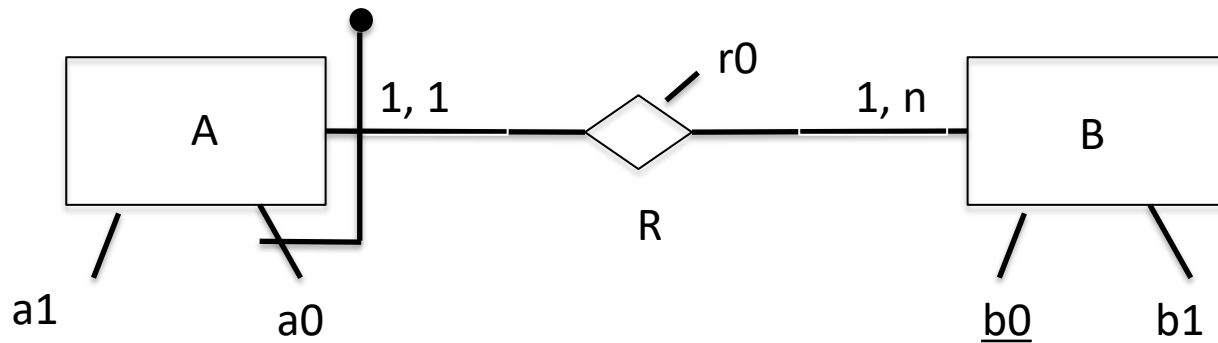


- $A(\underline{a_0}, a_1, b_0^*, r_0), B(\underline{b_0}, b_1)$  oppure
- $A(\underline{a_0}, a_1), B(\underline{b_0}, b_1, a_0^*, r_0)$



- $A(\underline{a_0}, a_1), B(\underline{b_0}, b_1, a_0^*, r_0)$  – soluzione preferita

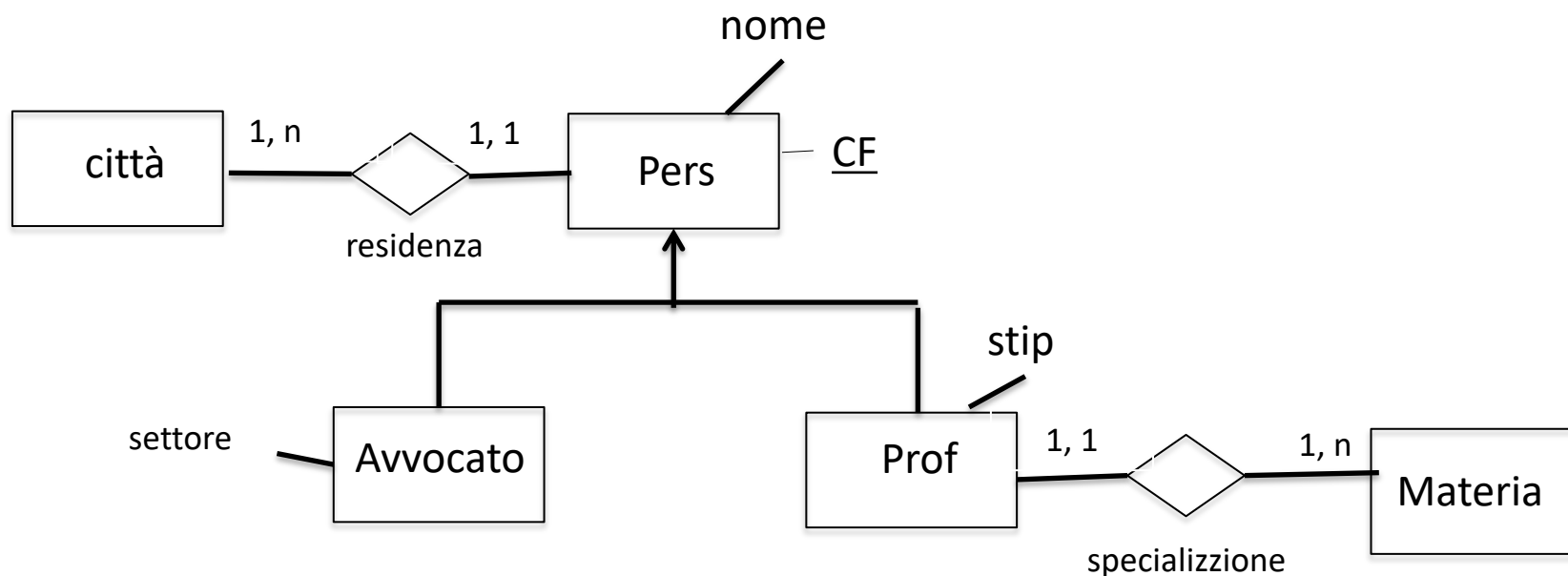
# Quadro riassuntivo



- $A(\underline{a0}, \underline{b0^*}, a1, r0)$
- $B(\underline{b0}, b1)$

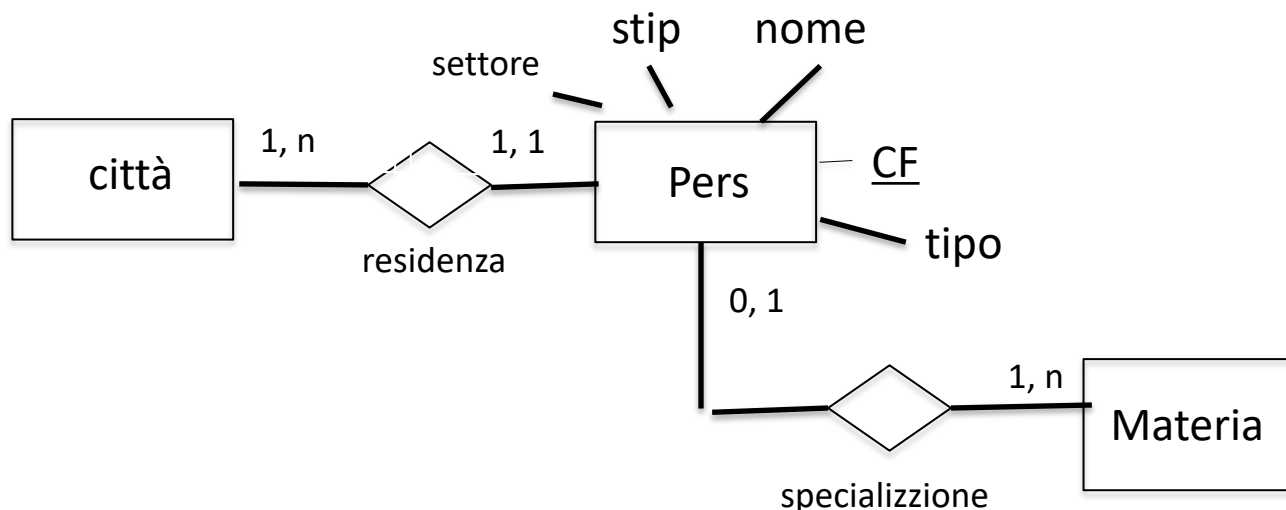
# Eliminazione delle Generalizzazioni

Prima di tradurre uno schema ER in uno schema relazionale, è necessario eliminare eventuali generalizzazioni, riconducendole ai costrutti di base del modello ER. Solo successivamente si procede alla traduzione seguendo le regole sopra esposte



# Eliminazione delle Generalizzazioni

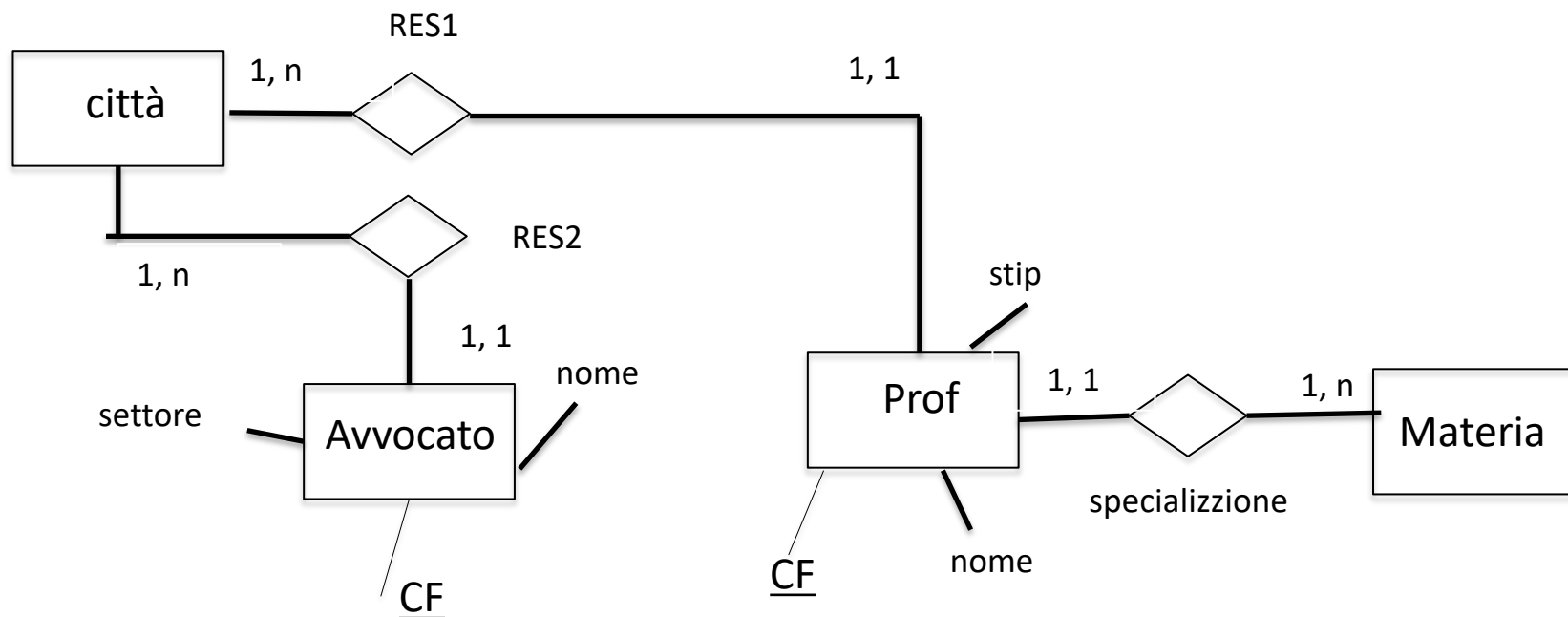
## 1) Accorpamento delle figlie nel genitore



- Scompaiono le entità figlie le cui istanze vengono immerse nel genitore
- L'associazione con *Materia* diventa opzionale

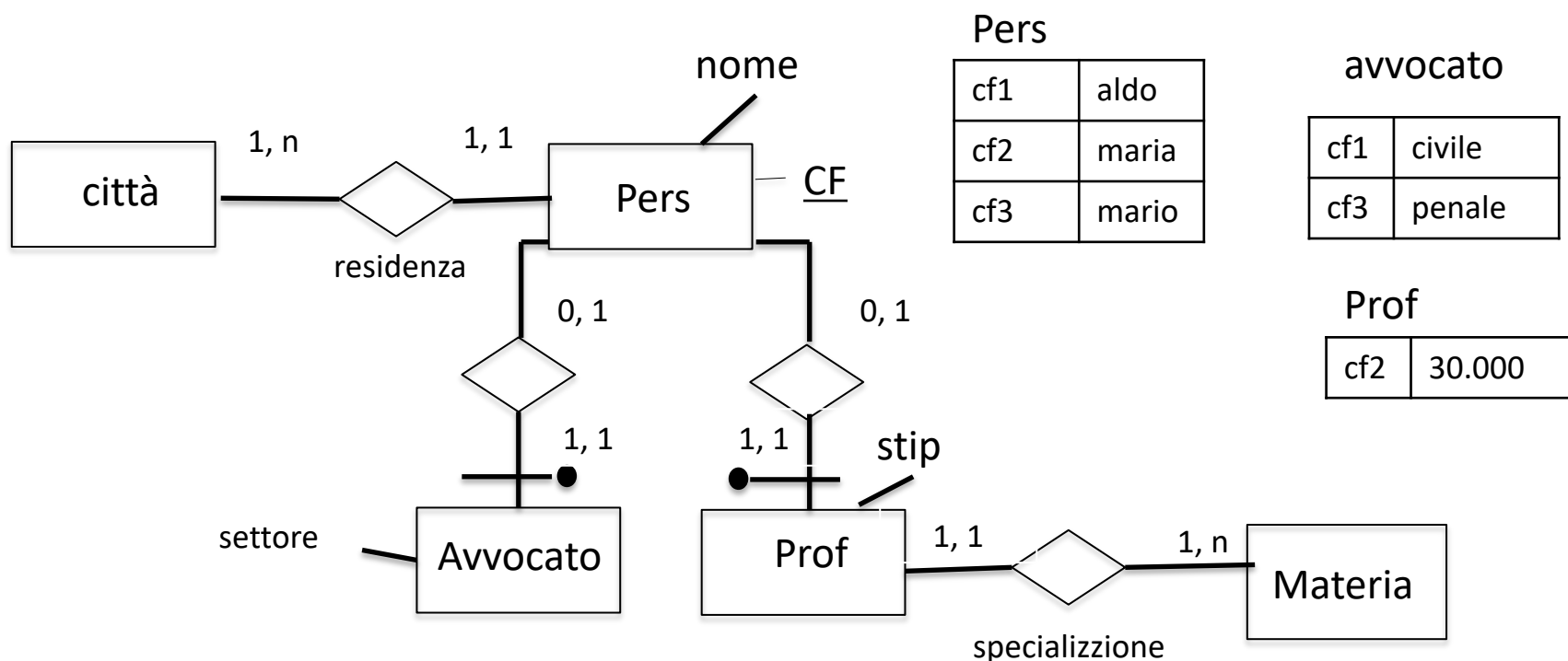
# Eliminazione delle Generalizzazioni

2) Accorpamento del genitore nelle figlie – *possibile solo se la generalizzazione è totale*



# Eliminazione delle Generalizzazioni

## 3) Sostituzione generalizzazione con associazioni

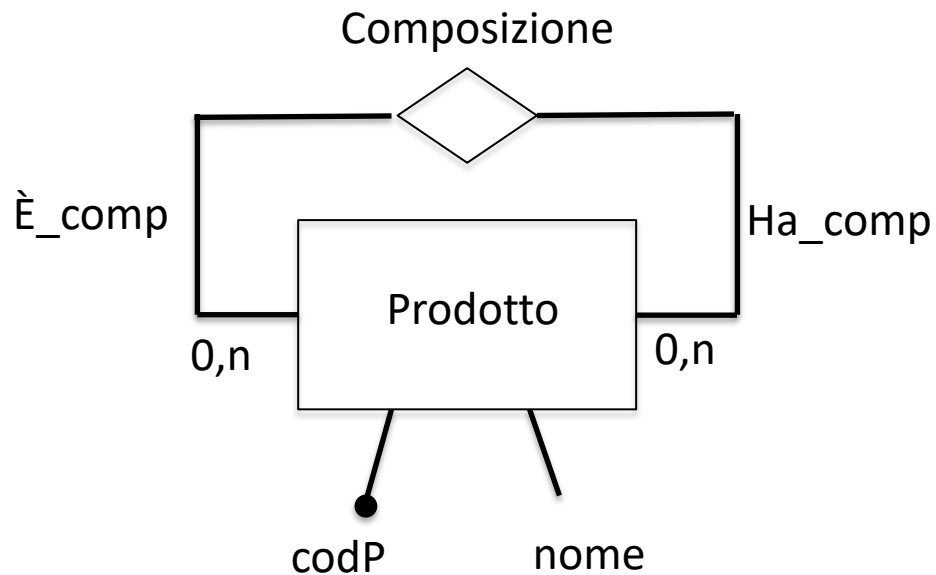


- Rimangono tutte le entità dello schema iniziale

# Esercizi proposti sulla Progettazione Logica

# ESERCIZIO 1

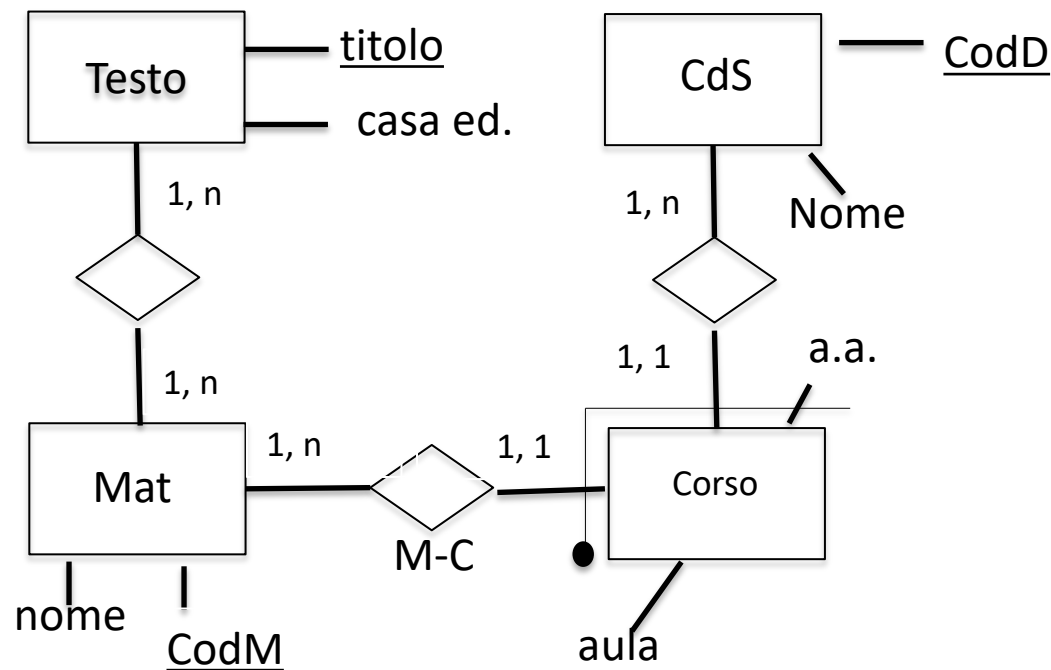
- Produrre lo schema relazionale equivalente al seguente schema ER





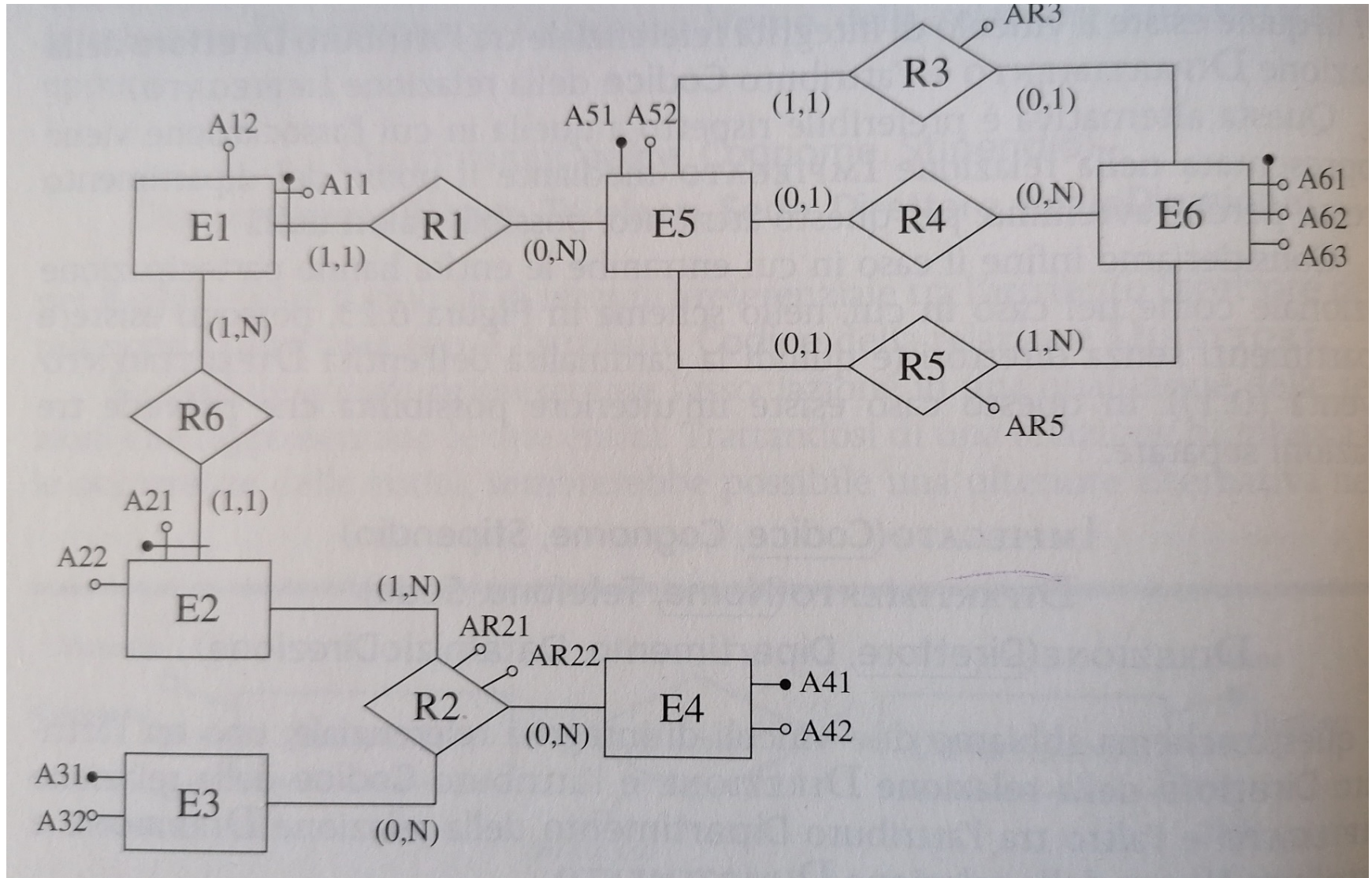
# ESERCIZIO 2

- Produrre uno schema relazionale equivalente al seguente schema ER



# ESERCIZIO 3

(pag. 306 Basi di Dati – P. Atzeni et al.)



# ESERCIZIO 4

- Si traduca il seguente schema ER in uno schema relazionale, previa eliminazione delle generalizzazioni. A tal fine si assuma che la generalizzazione Professionista sia totale e la generalizzazione Prof sia parziale

