

# Esame di Basi di Dati

## A.A. 2021/2022 – Appello del 21/01/2022

### Problema 1

Si richiede di progettare lo schema concettuale Entità-Relazione relativo al sistema informativo che tiene traccia delle mostre nel territorio italiano, in accordo ai seguenti requisiti. Sono di interesse le mostre allestite in Italia, con nome (identificativo), costo di allestimento ed argomento (arte, scienza, storia, ecc.). Ogni mostra può essere allestita in diverse città ed ogni volta che viene allestita ha una persona organizzatrice. Vigge la regola che una persona non può essere organizzatrice di più di una mostra nella stessa città. Per l'allestimento delle mostre è prevista la possibilità di usufruire di un contributo finanziario, ma solo nel caso in cui la persona organizzatrice sia un Premio Nobel: quando questo succede, è di interesse sapere l'ammontare del contributo. Di ogni città interessa il codice (identificativo) e la regione. Di ogni persona interessa il codice fiscale (identificativo) e la data di nascita. Di ogni persona insignita del Premio Nobel interessa anche l'anno dal quale ha questo titolo. Infine, interessa anche sapere chi sono le persone che hanno diretto le mostre nel tempo, sapendo che una mostra non può cambiare più di un direttore all'anno.

### Problema 2

Si richiede di effettuare la progettazione logica per il sistema informativo citato nel problema 1, tenendo conto dell'indicazione che quando si accede alle informazioni su un allestimento di mostre, si vuole sempre sapere se per esso è stato concesso un contributo ed in caso positivo, si vuole sempre conoscere l'ammontare del contributo stesso.

### Problema 3

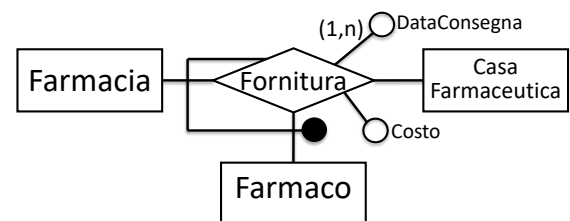
Si consideri una base di dati che comprende le relazioni **Azienda** e **Finanziamento**. La prima relazione ha come schema **Azienda**(codice, tipo, capitale) e memorizza il codice (chiave primaria), il tipo ed il capitale delle aziende. La seconda relazione ha come schema **Finanziamento**(istituto, azienda, ammontare) e memorizza le informazioni su quanto finanziamento (maggiore di 0) gli istituti concedono alle aziende. È noto che la base di dati soddisfa il vincolo di integrità referenziale da **Finanziamento**[azienda] ad **Azienda**[codice]. Scrivere una query in SQL che per ogni istituto e per ogni tipo di azienda calcoli la quantità totale del finanziamento concesso da quell'istituto alle aziende di quel tipo, mostrando l'istituto, il tipo di azienda e la quantità totale suddetta.

### Problema 4

Riferendosi ancora alla base di dati menzionata nel problema 3, scrivere una query in algebra relazionale che calcoli le aziende che hanno ottenuto un finanziamento da tutti gli istituti.

### Problema 5

Si consideri lo schema concettuale  $S_1$  mostrato qui a destra e si illustri lo schema concettuale ristrutturato che si ottiene eseguendo il passo di ristrutturazione di  $S_1$ .



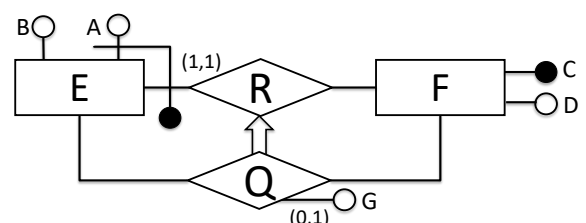
### Problema 6

Si consideri lo schema concettuale  $S_2$  mostrato qui a destra e si supponga che le indicazioni di progetto specifichino che ogni volta che si accede ad una istanza di **R** si vuole sempre sapere se essa è una istanza di **Q** e, in caso positivo, si vuole sempre sapere l'eventuale valore di **G** per quella istanza. Si supponga che il risultato del progetto sia il seguente schema logico  $L$

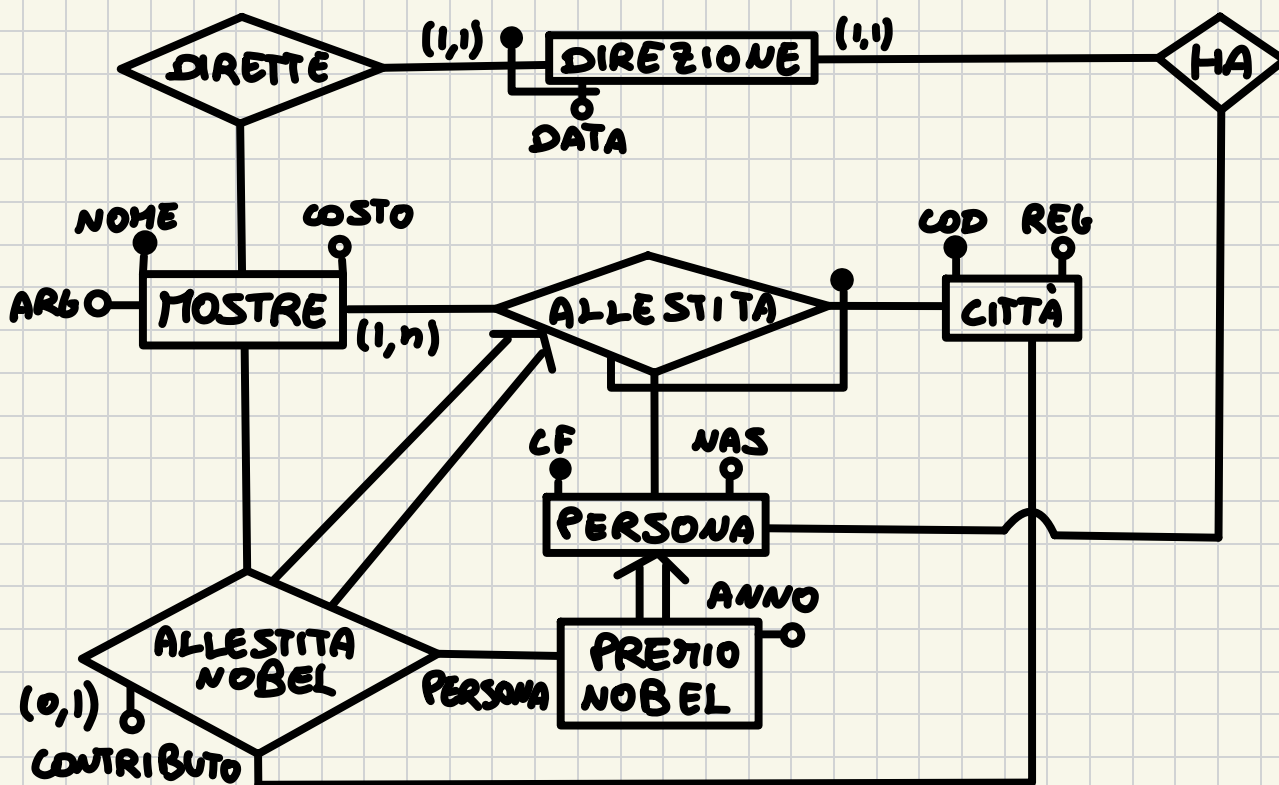
$F(\underline{C}, D)$

$E(\underline{A}, B, \underline{F}, G^*)$  foreign key  $E[F] \subseteq F[C]$

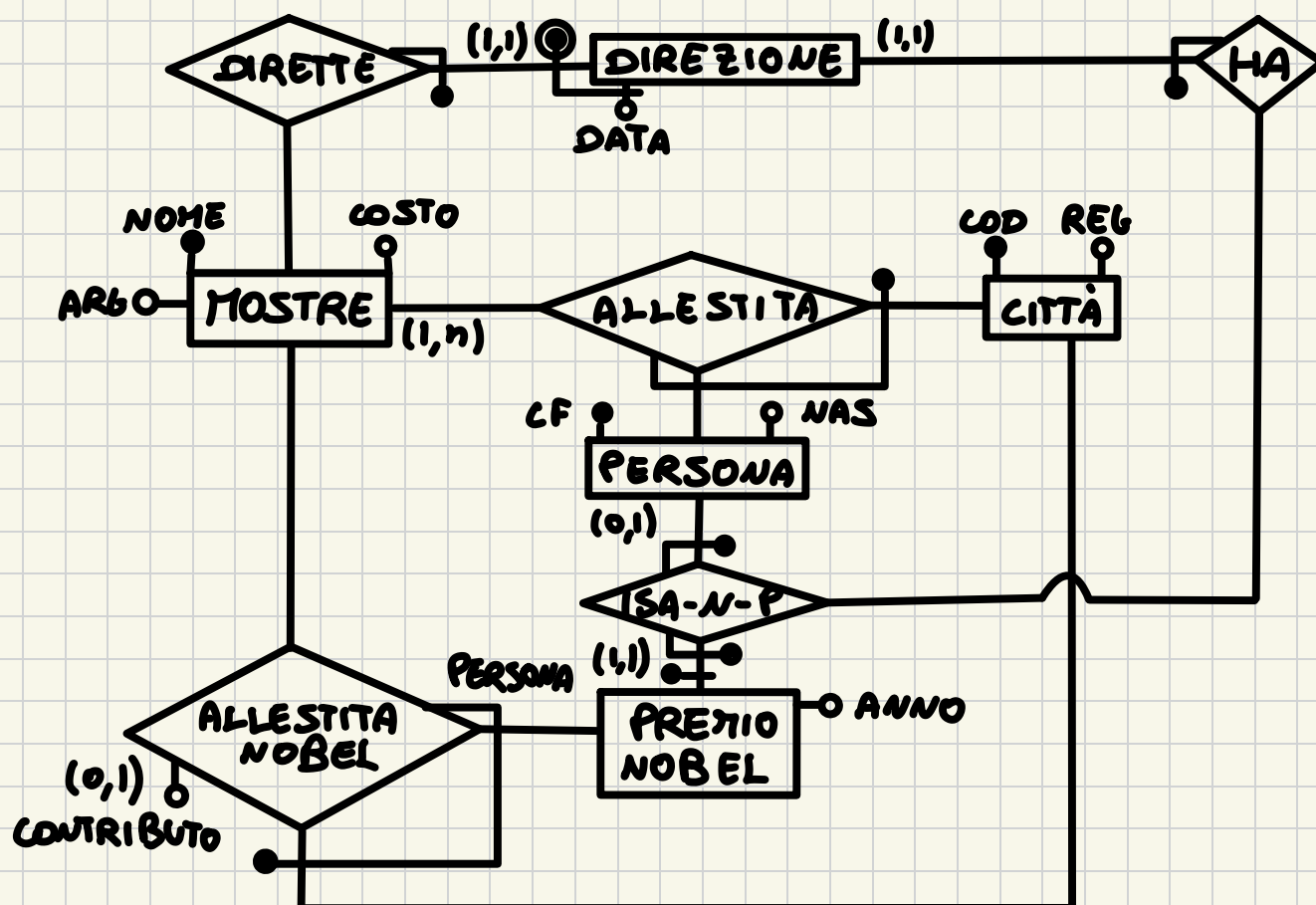
e si dica se tale schema  $L$  è una corretta traduzione di  $S_2$  in termini del modello relazionale. Se la risposta è positiva, motivare tale risposta; se invece è negativa, mostrare una istanza di  $S_2$  che non può essere correttamente rappresentata in  $L$ .



## PROBLEMA 1



## PROBLEMA 2



OGNI ISTANZA DI ALLNOBEL È ISTANZA DI ALLESTITA

MOSTRE (NOME, COSTO, ARG)

INCLUSIONE: MO[NOME]  $\subseteq$  ALLESTITA[MOSTRE]

CITTÀ (COD, REG)

PERSONA (CF, NAS)

PREMIO NOBEL (PERSONA, ANNO)

FOREIGN KEY: PN[P]  $\subseteq$  P[CF]

DIREZIONE (MOSTRE, DATA)

FOREIGN KEY: DIR[MOSTRE]  $\subseteq$  MOSTRE[NOME]

FOREIGN KEY: DIR[DATA, MOSTRA]  $\subseteq$  HA[DATA, MOSTRA]

HA (MOSTRA, DATA, PERSONA)

FOREIGN KEY: DI[M, D]  $\subseteq$  DIR[DATA, MOSTRA]

FOREIGN KEY: DI[P]  $\subseteq$  P[CF]

ALLESTITA (PERSONA, CITTÀ, MOSTRA)

FOREIGN KEY: A[P]  $\subseteq$  P[CF]

FOREIGN KEY: A[C]  $\subseteq$  C[COD]

FOREIGN KEY: A[N]  $\subseteq$  N[NOME]

ALLESTITA NOBEL (PREMIO NOBEL, CITTÀ, MOSTRA, CONTRIBUTO\*)

FOREIGN KEY: AN[PN]  $\subseteq$  PN[CF]

FOREIGN KEY: AN[C]  $\subseteq$  C[COD]

FOREIGN KEY: AN[MOS]  $\subseteq$  MOS[NOME]

FOREIGN KEY: AN[PN, C, MOS]  $\subseteq$  ALLESTITA[PN, C, MOS]

---

ALLESTITA (PERSONA, CITTÀ, MOSTRA, SUPPORTATA, CONTRIBUTO\*)

FOREIGN KEY: A[P]  $\subseteq$  P[CF]

FOREIGN KEY: A[C]  $\subseteq$  C[COD]

FOREIGN KEY: A[N]  $\subseteq$  N[NOME]

VINCOLO: SE SUPPORTATA=FALSE, CONTRIBUTO IS NULL

### PROBLEMA 3

```
SELECT DISTINCT F.ISTITUTO, A.TIPO, SUM (F.AMONTARE)
FROM AZIENDA A JOIN FINANZIAMENTO F ON F.AZIENDA=A.CODICE
GROUP BY F.ISTITUTO, A.TIPO

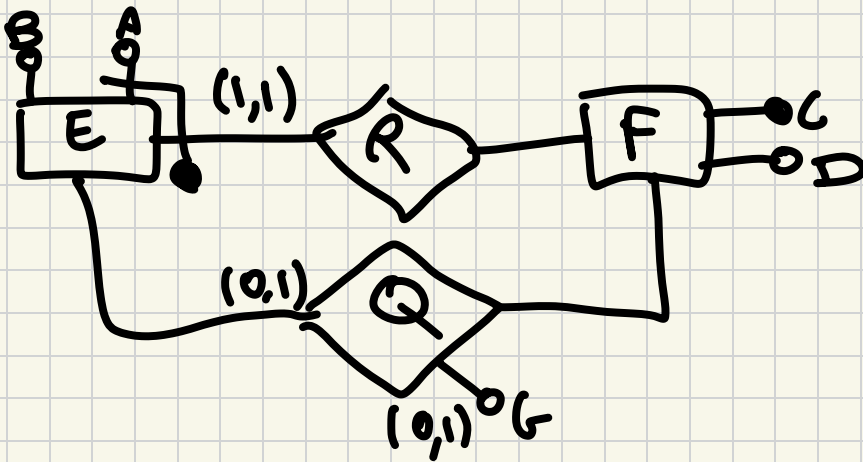
UNION
```

```
SELECT DISTINCT F.ISTITUTO, A.TIPO, 0 AS SOMMA
FROM AZIENDA A, FINANZIAMENTO F
WHERE (F.ISTITUTO, A.TIPO) NOT IN (SELECT N.ISTITUTO, Z.TIPO
FROM " " " "
ON N.AZIENDA=Z.CODICE)
```

### PROBLEMA 4

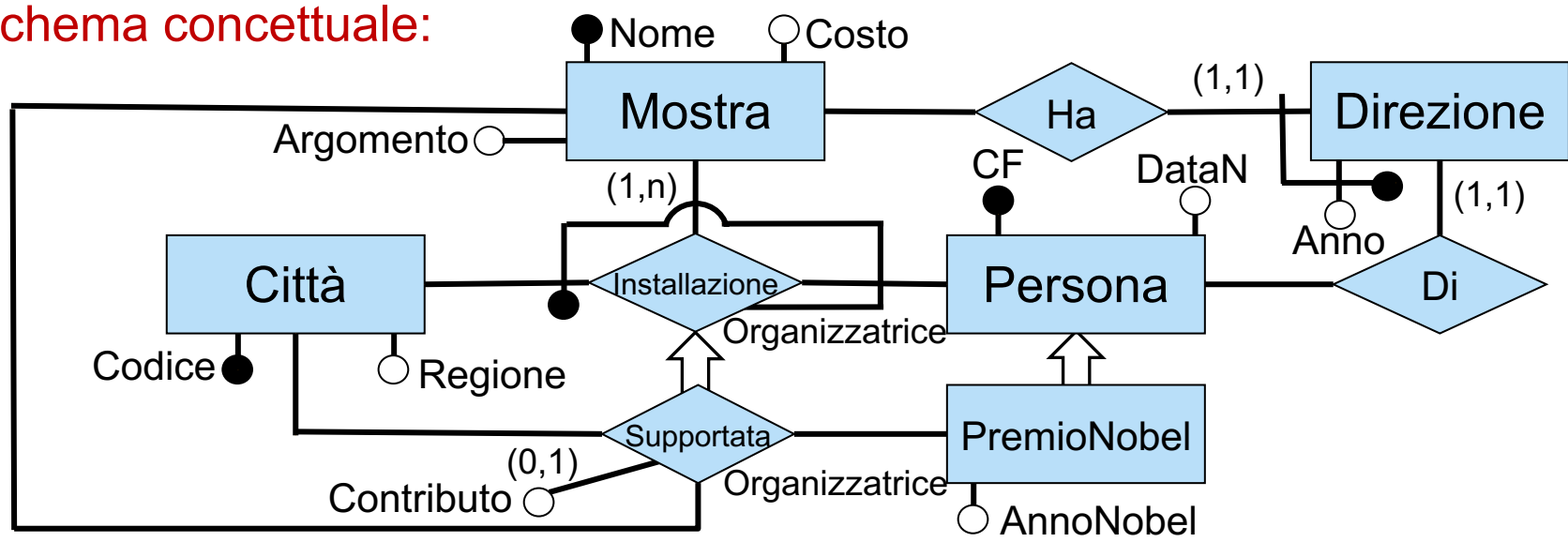
PROJ<sub>CODICE</sub> (AZIENDA)

PROJ<sub>CODICE</sub> ( PROJ<sub>CODICE, ISTITUTO</sub> (AZIENDA JOIN FINANZIAMENTO)  
- PROJ<sub>CODICE, ISTITUTO</sub> (REN<sub>CODICE ← AZIENDA</sub> (FINANZIAMENTO)))

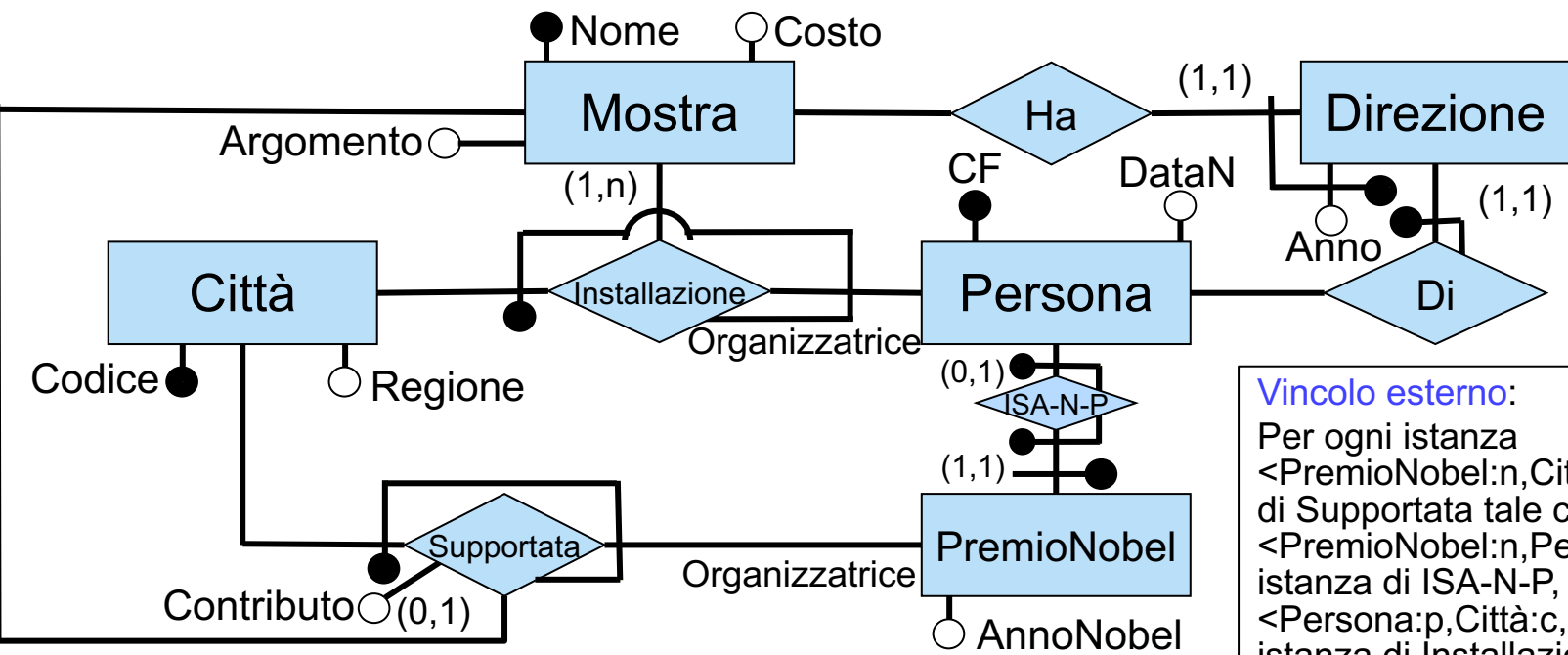


# Problema 1 e 2 – Schema ER e sua ristrutturazione

## Schema concettuale:



## Schema concettuale ristrutturato:



**Vincolo esterno:**  
Per ogni istanza  $\langle \text{PremioNobel:n}, \text{Città:c}, \text{Mostra:m} \rangle$  di Supportata tale che  $\langle \text{PremioNobel:n}, \text{Persona:p} \rangle$  è istanza di ISA-N-P,  $\langle \text{Persona:p}, \text{Città:c}, \text{Mostra:m} \rangle$  è istanza di Installazione.

# Problema 2 – Traduzione diretta e ristrutturazione schema logico

Mostra(nome, costo, argomento)

**inclusione:** Mostra[nome]  $\subseteq$  Installazione[mostra]

Persona(codicefiscale, datanascita)

PremioNobel(codicefiscale, annoNobel)

**foreign key:** PremioNobel[codicefiscale]  $\subseteq$  Persona[codicefiscale]

Città(codice, regione)

Installazione(organizzatrice, città, mostra)

**foreign key:** Installazione[mostra]  $\subseteq$  Mostra[nome]

**foreign key:** Installazione[organizzatrice]  $\subseteq$  Persona[codicefiscale]

**foreign key:** Installazione[città]  $\subseteq$  Città[codice]

Supportata(organizzatrice, città, mostra, contributo\*)

**foreign key:** Supportata[organizzatrice]  $\subseteq$  PremioNobel[codicefiscale]

**foreign key:** Supportata[organizzatrice, città, mostra]  $\subseteq$  Installazione[organizzatrice, città, mostra]

Direzione(mostra, data)

**foreign key:** Direzione[mostra, data]  $\subseteq$  Di[mostra, data]

**foreign key:** Direzione[mostra]  $\subseteq$  Mostra[nome]

Di(mostra, data, persona)

**foreign key:** Di[mostra, data]  $\subseteq$  Direzione[mostra, data]

**foreign key:** Di[persona]  $\subseteq$  Persona[codicefiscale]

Mostra(nome, costo, argomento)

**inclusione:** Mostra[nome]  $\subseteq$  Installazione[mostra]

Persona(codicefiscale, datanascita)

PremioNobel(codicefiscale, annoNobel)

**foreign key:** PremioNobel[codicefiscale]  $\subseteq$  Persona[codicefiscale]

Città(codice, regione)

Installazione(organizzatrice, città, mostra, supportata, contributo\*)

**foreign key:** Installazione[mostra]  $\subseteq$  Mostra[nome]

**foreign key:** Installazione[organizzatrice]  $\subseteq$  Persona[codicefiscale]

**foreign key:** Installazione[città]  $\subseteq$  Città[codice]

**vincolo di tupla:** se supportata = false allora contributo is null

**inclusione:** (select organizzatrice from Installazione where supportata = true)  $\subseteq$  PremioNobel[codicefiscale]

Direzione(mostra, data)

**foreign key:** Direzione[mostra, data]  $\subseteq$  Di[mostra, data]

**foreign key:** Direzione[mostra]  $\subseteq$  Mostra[nome]

Di(mostra, data, persona)

**foreign key:** Di[mostra, data]  $\subseteq$  Direzione[mostra, data]

**foreign key:** Di[persona]  $\subseteq$  Persona[codicefiscale]

Schema logico prodotto  
dalla traduzione diretta:

Schema logico  
ristrutturato:

View Supportata = (select organizzatore, città  
mostra, contributo  
from Installazione  
where supportata = true)

## Problema 3 – testo e soluzione

Si consideri una base di dati che comprende le relazioni Azienda e Finanziamento. La prima relazione ha come schema **Azienda(codice, tipo, capitale)** e memorizza il codice (chiave primaria), il tipo ed il capitale delle aziende. La seconda relazione ha come schema **Finanziamento(istituto, azienda, ammontare)** e memorizza le informazioni su quanto finanziamento gli istituti concedono alle aziende. È noto che la base di dati soddisfa il vincolo di integrità referenziale da **Finanziamento[azienda]** ad **Azienda[codice]**. Scrivere una query in SQL che per ogni istituto e per ogni tipo di azienda calcoli la quantità totale del finanziamento concesso da quell'istituto alle aziende di quel tipo, mostrando l'istituto, il tipo di azienda e la quantità totale suddetta.

### Soluzione:

```
select distinct f.istituto, a.tipo, 0 as totale
from Finanziamento f, Azienda a
where (f.istituto, a.tipo) not in (select g.istituto, b.tipo
                                from Finanziamento g join Azienda b
                                on b.codice=g.azienda)

union

select f.istituto, a.tipo, sum(f.ammontare) as totale
from Azienda a join Finanziamento f on a.codice = f.azienda
group by f.istituto, a.tipo
```

## Problema 4 – testo e soluzione

Riferendosi ancora alla base di dati menzionata nel problema 3, scrivere una query in algebra relazionale che calcoli le aziende che hanno ottenuto un finanziamento da tutti gli istituti.

### Soluzione:

Assumiamo che i valori che compaiono in ammontare siano positivi. Da tutte le azienda si tolgono quelle che non hanno ricevuto il finanziamento da almeno un istituto. Per calcolare le aziende che non hanno ricevuto il finanziamento da almeno un istituto prima si tolgono da tutte le coppie  $\langle a, i \rangle$  dove  $a$  è un'azienda e  $i$  è un istituto quelle tali che  $a$  ha avuto il finanziamento da  $i$ . Le coppie rimaste sono appunto le coppie  $\langle a, i \rangle$  tali che  $a$  non ha ricevuto il finanziamento da  $i$ . Se ora si calcola la proiezione di queste coppie sulla prima componente si ottengono proprio le aziende che non hanno ricevuto il finanziamento da almeno un istituto. La query in algebra relazionale è quindi:

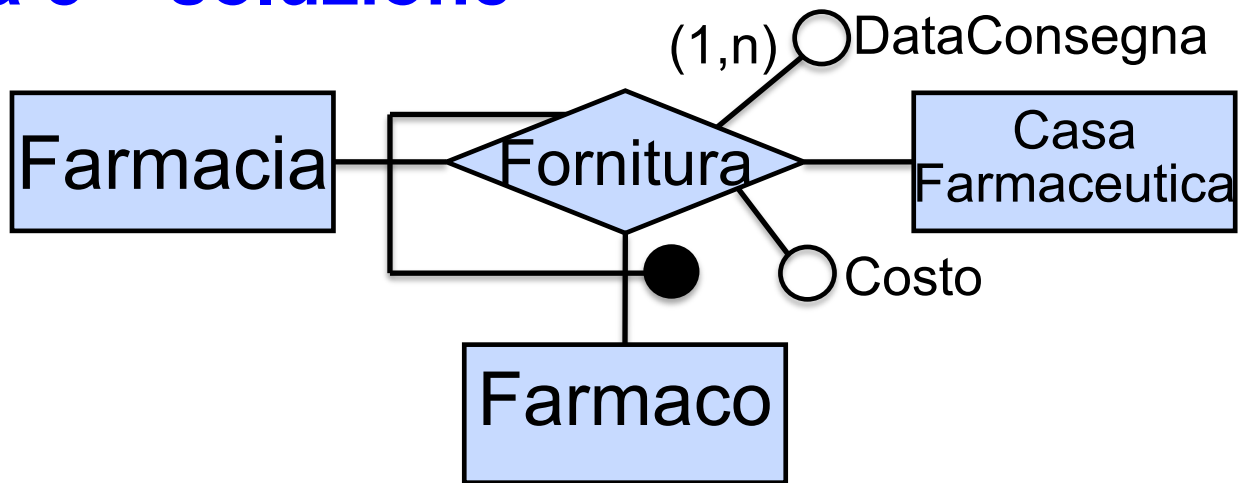
$\text{PROJ}_{\text{codice}}(\text{Azienda})$

-

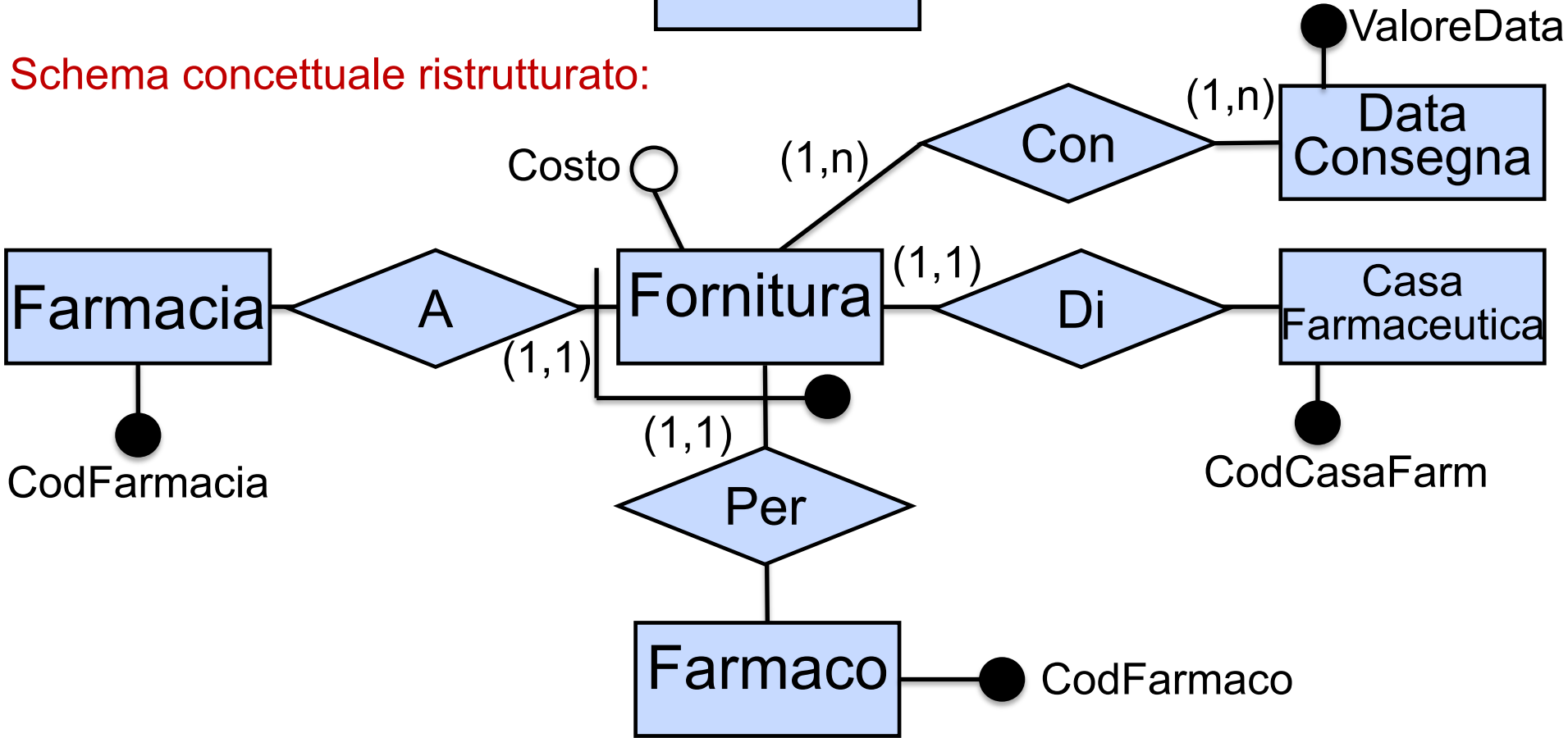
$\text{PROJ}_{\text{codice}}(\text{PROJ}_{\text{codice,istituto}}(\text{Azienda JOIN Finanziamento}) - \text{PROJ}_{\text{codice,istituto}}(\text{REN}_{\text{codice} \leftarrow \text{azienda}}(\text{Finanziamento})))$



# Problema 5 – soluzione

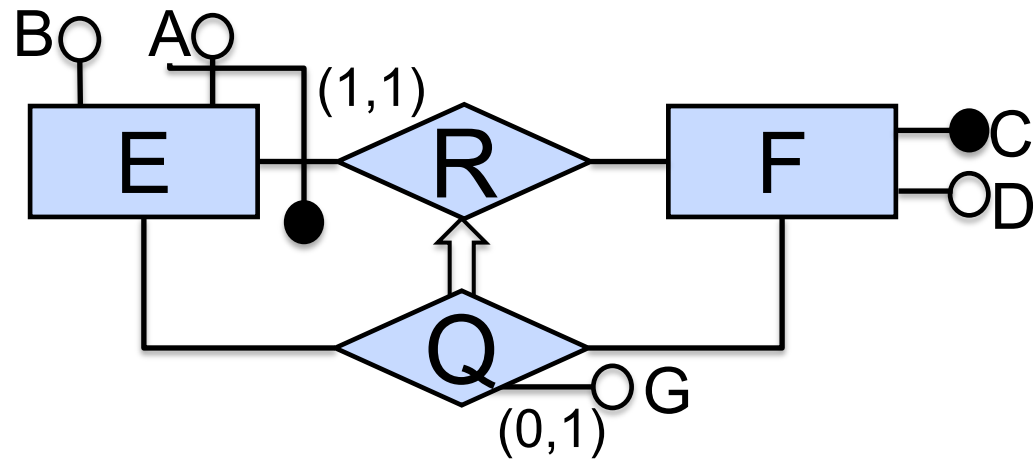


Schema concettuale ristrutturato:



# Problema 6 – soluzione

Schema  $S_2$ :



Lo schema logico L:

$F(C,D)$

$E(A,B,F,G^*)$  foreign key  $E[F] \subseteq F[C]$

non è una corretta traduzione di  $S_2$  nel modello relazionale. Basta considerare che l'istanza I di  $S_2$  tale che:

Istanza(I,E)={e}, Istanza(I,F)={f}, Istanza(I,R)={<E:e,F:f>}, Istanza(I,Q)={<E:e,F:f>},  
 Istanza(I,A)={<e,a>}, Istanza(I,B)={<e,b>}, Istanza(I,C)={<f,c>}, Istanza(I,D)={<f,d>},  
 Istanza(I,G) = { }

non può essere correttamente rappresentata in L. Infatti, la traduzione di I in L è:

E	A	B	F	G	F	C	D
	a	b	c	null		c	d

che è la stessa della seguente istanza J (diversa da I) di  $S_2$ :

Istanza(J,E)={e}, Istanza(J,F)={f}, Istanza(J,R)={<E:e,F:f>}, Istanza(J,Q)= { },  
 Istanza(J,G) = { }, Istanza(J,A)={<e,a>}, Istanza(J,B)={<e,b>}, Istanza(J,C)={<f,c>},  
 Istanza(J,D)={<f,d>}

Questo dimostra che lo schema L non riesce a discriminare istanze diverse di  $S_2$ .