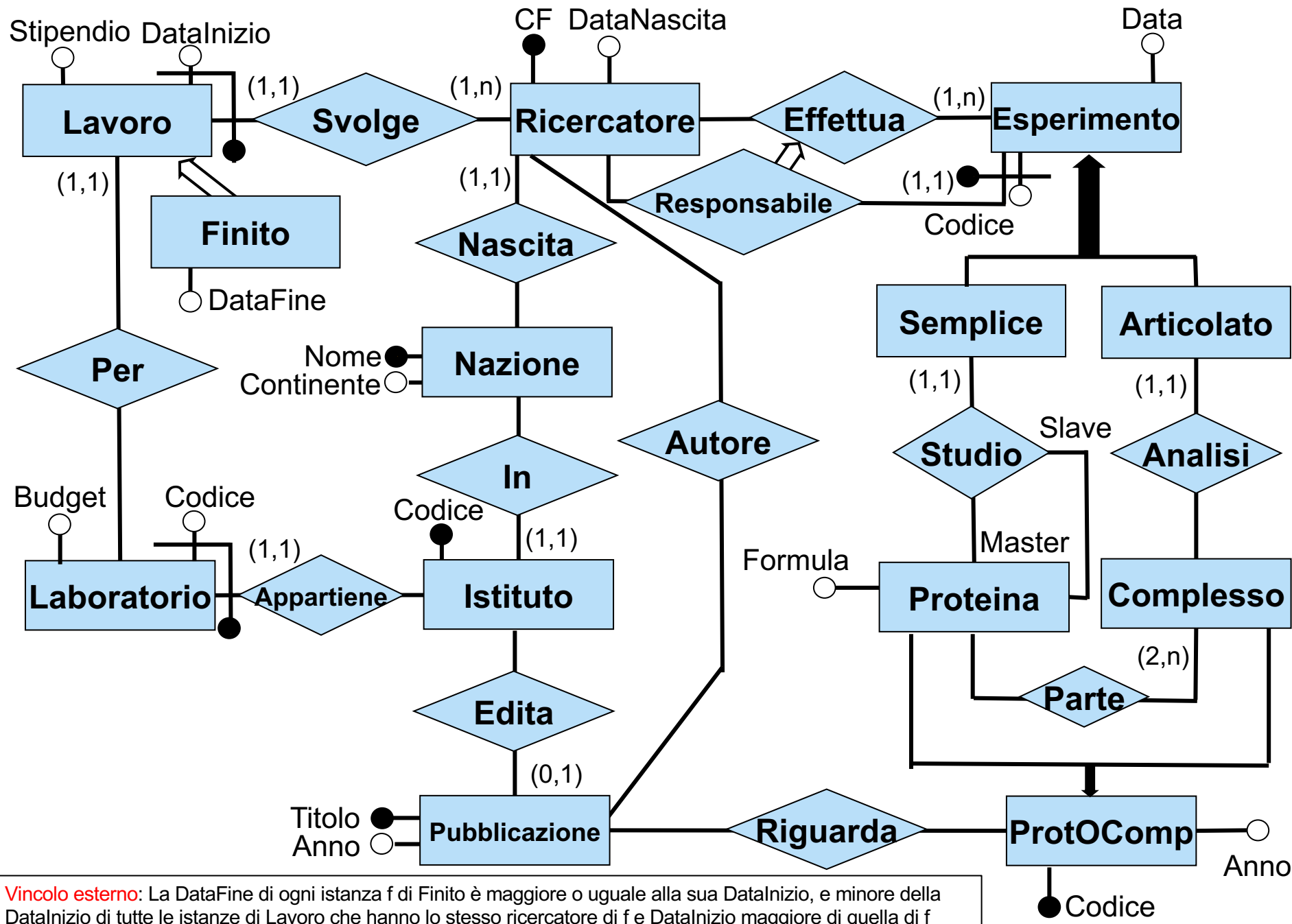


Basi di dati

Soluzione per l'appello del 20-02-2019

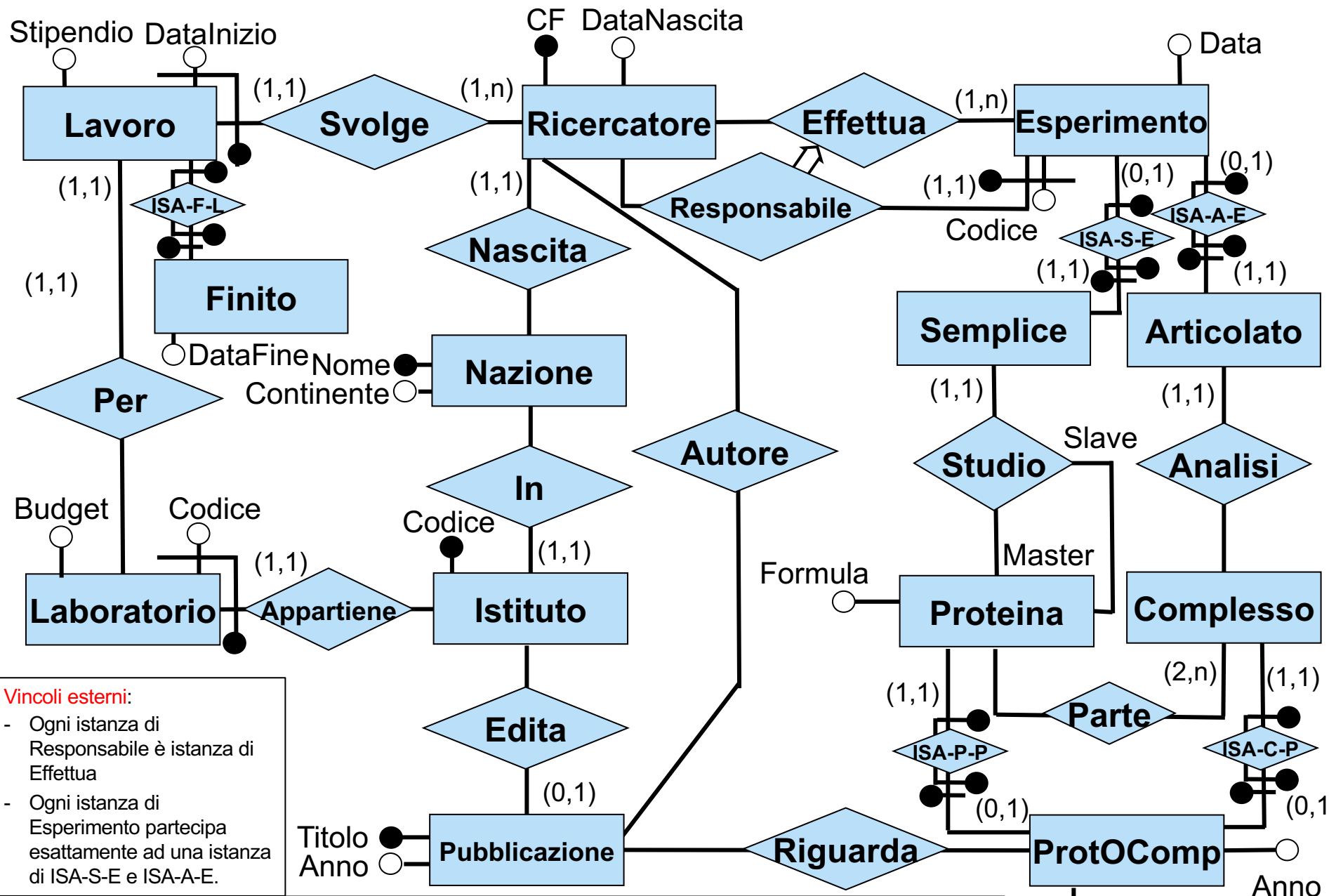
Anno Accademico 2018/19

Problema 1 – Schema concettuale



Vincolo esterno: La DataFine di ogni istanza f di Finito è maggiore o uguale alla sua DataInizio, e minore della DataInizio di tutte le istanze di Lavoro che hanno lo stesso ricercatore di f e DataInizio maggiore di quella di f

Problema 1 – Schema concettuale



- Vincoli esterni:**
- Ogni istanza di Responsabile è istanza di Effettua
 - Ogni istanza di Esperimento partecipa esattamente ad una istanza di ISA-S-E e ISA-A-E.

Vincolo esterno: La DataFine di ogni istanza f di Finito è maggiore o uguale alla sua DataInizio, e minore della DataInizio di tutte le istanze di Lavoro che hanno lo stesso ricercatore di f e DataInizio maggiore di quella di f

● Codice

Problema 2 - Schema logico dopo la traduzione diretta

Ricercatore(CF, DataNascita)

inclusione: Ricercatore[CF] \subseteq Lavoro[CF]

foreign key: Ricercatore[CF] \subseteq Nascita[CF]

Nascita(CF, Nazione)

foreign key: Nascita[CF] \subseteq Ricercatore[CF]

foreign key: Nascita[Nazione] \subseteq Nazione[Nome]

Lavoro(CF, DataInizio, Stipendio)

foreign key: Lavoro[CF] \subseteq Ricercatore[CF]

foreign key: Lavoro[CF,DataInizio] \subseteq Per[CF,DataInizio]

Finito(CF, DataInizio, DataFine)

foreign key: Finito[CF,DataInizio] \subseteq Lavoro[CF,DataInizio]

vincolo di tupla: DataFine \geq DataInizio

Per(CF, DataInizio, CodLab, NazLab)

foreign key: Per[CF,DataInizio] \subseteq Lavoro[CF,DataInizio]

foreign key: Per[CodLab,NazLab] \subseteq Laboratorio[Codice,Nazione]

Laboratorio(Codice, Istituto, Budget)

foreign key: Laboratorio[Nazione] \subseteq Nazione[Nome]

Nazione(Nome, Continente)

Istituto(Codice)

foreign key: Scoperta[CodLab,NazLab] \subseteq Laboratorio[Codice,Nazione]

foreign key: Istituto[Codice] \subseteq In[Istituto]

In(Istituto, Nazione)

foreign key: In[Istituto] \subseteq Istituto[Codice]

foreign key: In[Nazione] \subseteq Nazione[Nome]

Edita(Pubblicazione, Istituto)

foreign key: Edita[Istituto] \subseteq Istituto[Codice]

foreign key: Edita[Pubblicazione] \subseteq Pubblicazione[Titolo]

Pubblicazione(Titolo, Anno)

Autore(Pubblicazione, Ricercatore)

foreign key: Autore[Ricercatore] \subseteq Ricercatore[CF]

foreign key: Autore[Pubblicazione] \subseteq Pubblicazione[Titolo]

Riguarda(Pubblicazione, ProtOComp)

foreign key: Riguarda[Pubblicazione] \subseteq Pubblicazione[Titolo]

foreign key: Riguarda[ProtOComp] \subseteq ProtOComp[Codice]

Problema 2 - Schema logico dopo la traduzione diretta

ProtOComp(Codice, Anno)

Proteina(Codice, Formula)

foreign key: Proteina[Codice] \subseteq ProtOComp[Codice]

vincolo esterno: Proteina[Codice] \cap Complesso[Codice] = \emptyset

Complesso(Codice)

foreign key: Complesso[Codice] \subseteq ProtOComp[Codice]

vincolo esterno inter-relazionale: (select count(*) from Parte where CodComplesso=Codice) > 1

Parte(Complesso, Proteina)

foreign key: Parte[Complesso] \subseteq Complesso[Codice]

foreign key: Parte[Proteina] \subseteq Proteina[Codice]

Studio(CodSemplice, Responsabile, Master, Slave)

foreign key: Studio[CodSemplice, Responsabile] \subseteq Semplice[Codice, Responsabile]

foreign key: Studio[Master] \subseteq Proteina[Codice]

foreign key: Studio[Slave] \subseteq Proteina[Codice]

Analisi(CodArticolato, Responsabile, CodComplesso)

foreign key: Analisi[CodArticolato, Responsabile] \subseteq Articolato[Codice, Responsabile]

foreign key: Analisi[Complesso] \subseteq Complesso[Codice]

Semplice(Codice, Responsabile)

foreign key: Semplice[Codice, Responsabile] \subseteq Esperimento[Codice, Responsabile]

foreign key: Semplice[Codice, Responsabile] \subseteq Studio[CodSemplice, Responsabile]

Articolato(Codice, Responsabile)

foreign key: Articolato[Codice, Responsabile] \subseteq Esperimento[Codice, Responsabile]

foreign key: Articolato[Codice, Responsabile] \subseteq Analisi[CodArticolato, Responsabile]

vincolo esterno: Articolato[Codice, Responsabile] \cap Semplice[Codice, Responsabile] = \emptyset

Esperimento(Codice, Responsabile, Data)

foreign key: Esperimento[Responsabile] \subseteq Ricercatore[CF]

inclusione: Esperimento[Codice, Responsabile] \subseteq Effettua[CodEsperimento, Responsabile]

vincolo esterno: Esperimento[Codice, Responsabile] \subseteq Semplice[Codice, Responsabile] \cup
Articolato[Codice, Responsabile]

Effettua(CodEsperimento, Responsabile, Ricercatore)

foreign key: Effettua[Ricercatore] \subseteq Ricercatore[CF]

foreign key: Effettua[CodEsperimento, Responsabile] \subseteq Esperimento[Codice, Responsabile]

Vincolo esterno:

Per ogni tupla $t = (x, y, d)$ di Esperimento esiste la tupla (y, x, y) in Effettua

(ovvero: il responsabile di un esperimento è tra i ricercatori che lo effettuano).

Problema 2 – Ristrutturazione dello schema logico

- *Indicazione 1:* quando si accede ai dati relativi ad un esperimento interessa sempre sapere il tipo dell'esperimento e l'oggetto di indagine, ovvero le proteine "master" e "slave" nel caso di esperimento semplice ed il complesso proteico nel caso di esperimento articolato;
 - Accorpamento di Esperimento con Semplice, Articolato, Studio e Analisi, con l'aggiunta del campo "Tipo", compresi tutti i vincoli necessari
- *Indicazione 2:* quando si accede ai dati relativi ad una pubblicazione interessa sempre sapere l'eventuale istituto che ne è l'editore.
 - Accorpamento di Pubblicazione con Edita, mediante foreign key che può assumere valore nullo.

Problema 2 - Schema logico dopo la ristrutturazione

Ricercatore(CF, DataNascita)

inclusione: Ricercatore[CF] \subseteq Lavoro[CF]

foreign key: Ricercatore[CF] \subseteq Nascita[CF]

Nascita(CF, Nazione)

foreign key: Nascita[CF] \subseteq Ricercatore[CF]

foreign key: Nascita[Nazione] \subseteq Nazione[Nome]

Lavoro(CF, DataInizio, Stipendio)

foreign key: Lavoro[CF] \subseteq Ricercatore[CF]

foreign key: Lavoro[CF,DataInizio] \subseteq Per[CF,DataInizio]

Finito(CF, DataInizio, DataFine)

foreign key: Finito[CF,DataInizio] \subseteq Lavoro[CF,DataInizio]

vincolo di tupla: DataFine \geq DataInizio

Per(CF, DataInizio, CodLab, NazLab)

foreign key: Per[CF,DataInizio] \subseteq Lavoro[CF,DataInizio]

foreign key: Per[CodLab,NazLab] \subseteq Laboratorio[Codice,Nazione]

Laboratorio(Codice, Istituto, Budget)

foreign key: Laboratorio[Nazione] \subseteq Nazione[Nome]

Nazione(Nome, Continente)

Istituto(Codice)

foreign key: Scoperta[CodLab,NazLab] \subseteq Laboratorio[Codice,Nazione]

foreign key: Istituto[Codice] \subseteq In[Istituto]

In(Istituto, Nazione)

foreign key: In[Istituto] \subseteq Istituto[Codice]

foreign key: In[Nazione] \subseteq Nazione[Nome]

Pubblicazione(Titolo, Anno, Istituto*)

foreign key: Pubblicazione[Istituto] \subseteq Istituto[Codice]

Autore(Pubblicazione, Ricercatore)

foreign key: Autore[Ricercatore] \subseteq Ricercatore[CF]

foreign key: Autore[Pubblicazione] \subseteq Pubblicazione[Titolo]

Riguarda(Pubblicazione, ProtOComp)

foreign key: Riguarda[Pubblicazione] \subseteq Pubblicazione[Titolo]

foreign key: Riguarda[ProtOComp] \subseteq ProtOComp[Codice]

Problema 2 - Schema logico dopo la ristrutturazione

ProtOComp(Codice, Anno)

Proteina(Codice, Formula)

foreign key: Proteina[Codice] \subseteq ProtOComp[Codice]

vincolo esterno: Proteina[Codice] \cap Complesso[Codice] = \emptyset

Complesso(Codice)

foreign key: Complesso[Codice] \subseteq ProtOComp[Codice]

vincolo esterno inter-relazionale: (select count(*) from Parte where CodComplesso=Codice) > 1

Parte(Complesso, Proteina)

foreign key: Parte[Complesso] \subseteq Complesso[Codice]

foreign key: Parte[Proteina] \subseteq Proteina[Codice]

Esperimento(Codice, Responsabile, Data, Tipo, Master*, Slave*, CodComplesso*)

foreign key: Esperimento[CodComplesso] \subseteq Complesso[Codice]

foreign key: Esperimento[Master] \subseteq Proteina[Codice]

foreign key: Esperimento[Slave] \subseteq Proteina[Codice]

foreign key: Esperimento[Responsabile] \subseteq Ricercatore[CF]

inclusione: Esperimento[Codice,Responsabile] \subseteq Effettua[CodEsperimento,Responsabile]

vincolo di dominio: Tipo = 'semplice' OR Tipo = 'articolato'

vincolo di tupla: CodComplesso non è null se e solo se Tipo = 'articolato'

vincolo di tupla: Master non è null se e solo se Tipo = 'semplice'

vincolo di tupla: Slave non è null se e solo se Tipo = 'semplice'

Effettua(CodEsperimento, Responsabile, Ricercatore)

foreign key: Effettua[Ricercatore] \subseteq Ricercatore[CF]

foreign key: Effettua[CodEsperimento,Responsabile] \subseteq Esperimento[Codice,Responsabile]

Vincolo esterno:

Per ogni tupla $t = (x,y,d)$ di Esperimento esiste la tupla (y,x,y) in Effettua (ovvero: il responsabile di un esperimento è tra i ricercatori che lo effettuano).

Problema 2 - Schema logico dopo la ristrutturazione

Le viste per ricostruire le relazioni che sono state eliminate dalla ristrutturazione sono:

Vista Semplice(Codice, Responsabile) =
(select Codice, Responsabile from Esperimento where Tipo = 'semplice')

Vista Articolato(Codice, Responsabile) =
(select Codice, Responsabile from Esperimento where Tipo = 'articolato')

Vista Studio(Codice, Responsabile, Master, Slave) =
(select Codice, Responsabile, Master, Slave
from Esperimento
where Tipo = 'semplice')

Vista Analisi(Codice, Responsabile, CodComplesso) =
(select Codice, Responsabile, CodComplesso
from Esperimento
where Tipo = 'articolato')

Vista Edita(Pubblicazione, Istituto) =
(select Titolo, Istituto from Pubblicazione)

Problema 3: soluzione

```
select t.citta, t.codScuola
from (select citta, codScuola, avg(voto) as uscita
      from Diplomato natural join Scuola
      group by citta, codScuola) t
where t.uscita >= all (select avg(voto)
                     from Diplomato natural join Scuola
                     where citta = t.citta
                     group by codScuola)
```

Problema 4: soluzione

Occorre togliere dalle scuole che hanno almeno un diplomato con voto 100 ogni scuola s che è in una città che ha una scuola diversa da s con almeno un diplomato con voto uguale da 100. Osserviamo che:

- le scuole che hanno almeno un diplomato con voto 100 sono caratterizzate dalla query (si noti il join naturale): $SEL_{voto=100}(\text{Diplomato}) \text{ JOIN Scuola}$
- le scuole con almeno un diplomato con voto diverso da 100 dalla query (anch'essa con il join naturale): $SEL_{voto \neq 100}(\text{Diplomato}) \text{ JOIN Scuola}$

Per calcolare le scuole s che sono in una città con una scuola diversa da s che ha almeno un diplomato con voto uguale a 100 basta proiettare su codScuola la query $\text{Scuola JOIN}_{citta=c \text{ AND } \text{codScuola} \neq s} (\text{REN}_{c \leftarrow citta, s \leftarrow \text{codScuola}} (SEL_{voto=100}(\text{Diplomato}) \text{ JOIN Scuola}))$

Ne segue che la query complessiva è:

$PROJ_{\text{codScuola}}(SEL_{voto=100}(\text{Diplomato}) \text{ JOIN Scuola})$

—

$PROJ_{\text{codScuola}}(\text{Scuola JOIN}_{citta=c \text{ and } \text{codScuola} \neq s} (\text{REN}_{c \leftarrow citta, s \leftarrow \text{codScuola}} (SEL_{voto=100}(\text{Diplomato}) \text{ JOIN Scuola})))$

Problema 5: soluzione

Non esiste una istanza dello schema S in cui l'entità $F1$ ha almeno una istanza. Dimostriamo questa affermazione per assurdo.

Supponiamo che esista una istanza I tale che $Istanze(I, F1)$ comprenda f .

Siccome $F1$ partecipa alla relazione $R2$ nel ruolo F con cardinalità $(1, n)$, deve esistere almeno una istanza $\langle E:e, F:f \rangle$ in $Istanze(I, R2)$ e, poiché $(R2 \text{ ISA } R)$, $\langle E:e, F:f \rangle$ è anche in $Istanze(I, R)$.

Siccome E partecipa alla relazione $R1$ nel ruolo E con cardinalità $(1, n)$, deve esistere almeno una istanza in $Istanze(I, R1)$ a cui partecipa e . Poiché $R1$ è disgiunto da $R2$ (generalizzazione completa), avremo una coppia $\langle E:e, F:g \rangle$ in $Istanze(I, R1)$, con g diverso da f , e poiché $(R1 \text{ ISA } R)$ abbiamo anche $\langle E:e, F:g \rangle$ in $Istanze(I, R)$.

A questo punto, però, in I abbiamo le due istanze $\langle E:e, F:f \rangle$ e $\langle E:e, F:g \rangle$ di R a cui la stessa istanza e di E partecipa, e questo viola il vincolo di cardinalità $(1, 1)$ di E rispetto a R . Ciò significa che I non è una istanza lecita di S .

Abbiamo quindi dimostrato che assumere l'esistenza di una istanza I dello schema S in cui l'entità $F1$ ha almeno una istanza ci porta a negare che I sia una istanza di S . Quindi, concludiamo che una tale istanza non esiste.