

# Basi di dati

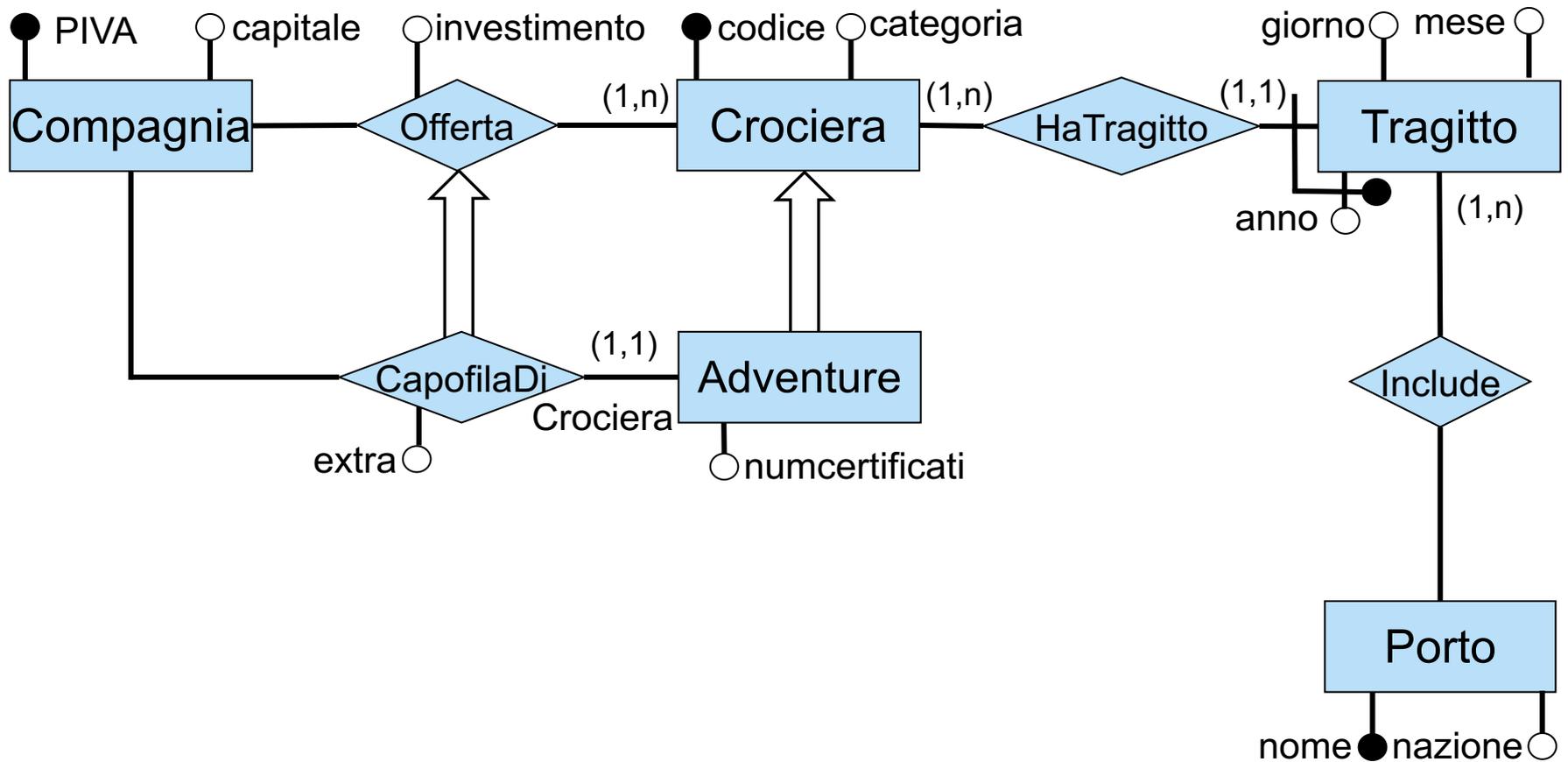
**Soluzioni dei problemi proposti  
nell'appello del 26-01-2024  
Compito A**

***Maurizio Lenzerini***

Anno Accademico 2023/24

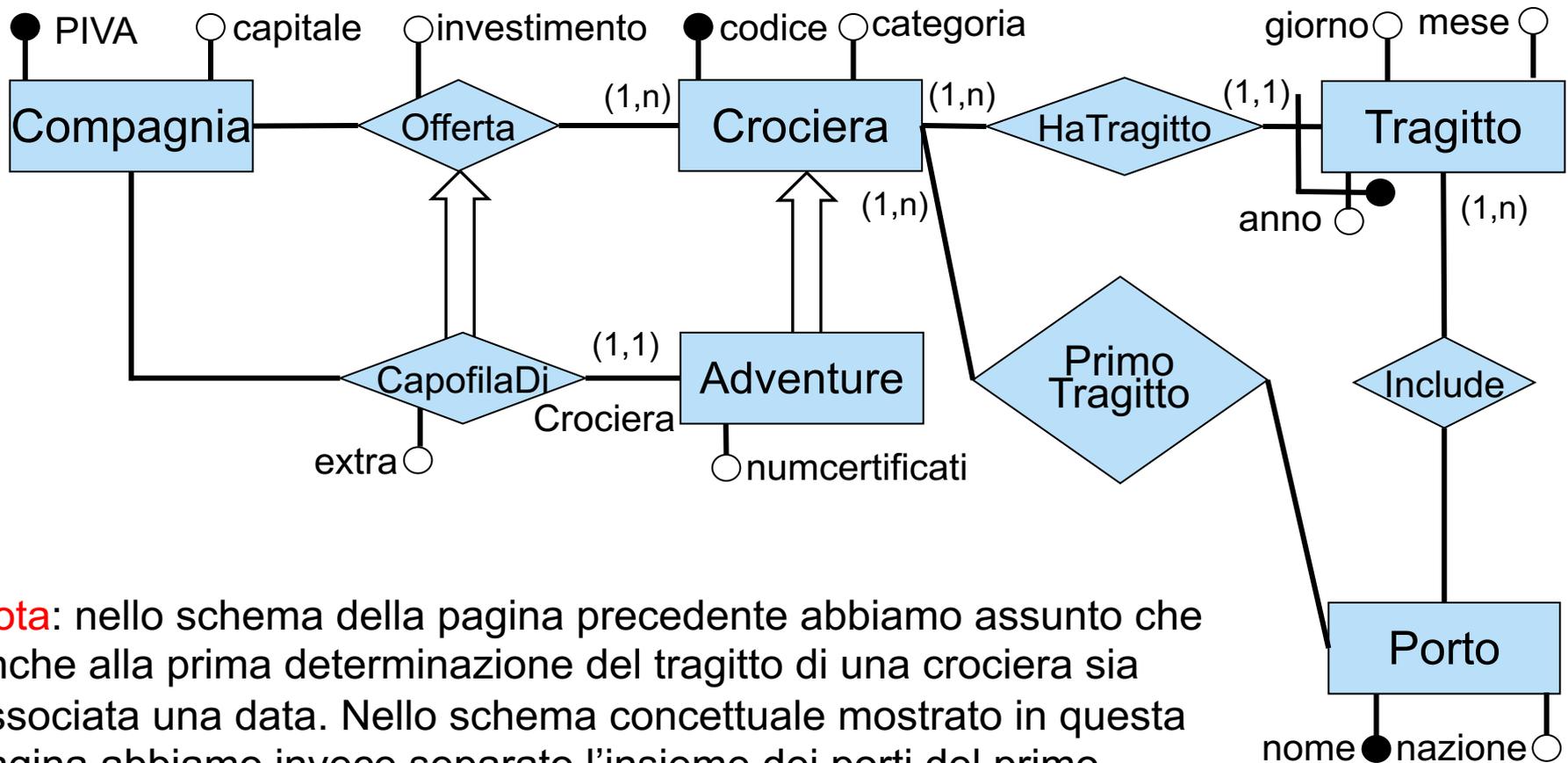
# Problema 1 – Schema ER

Schema concettuale:



# Problema 1 – Osservazione

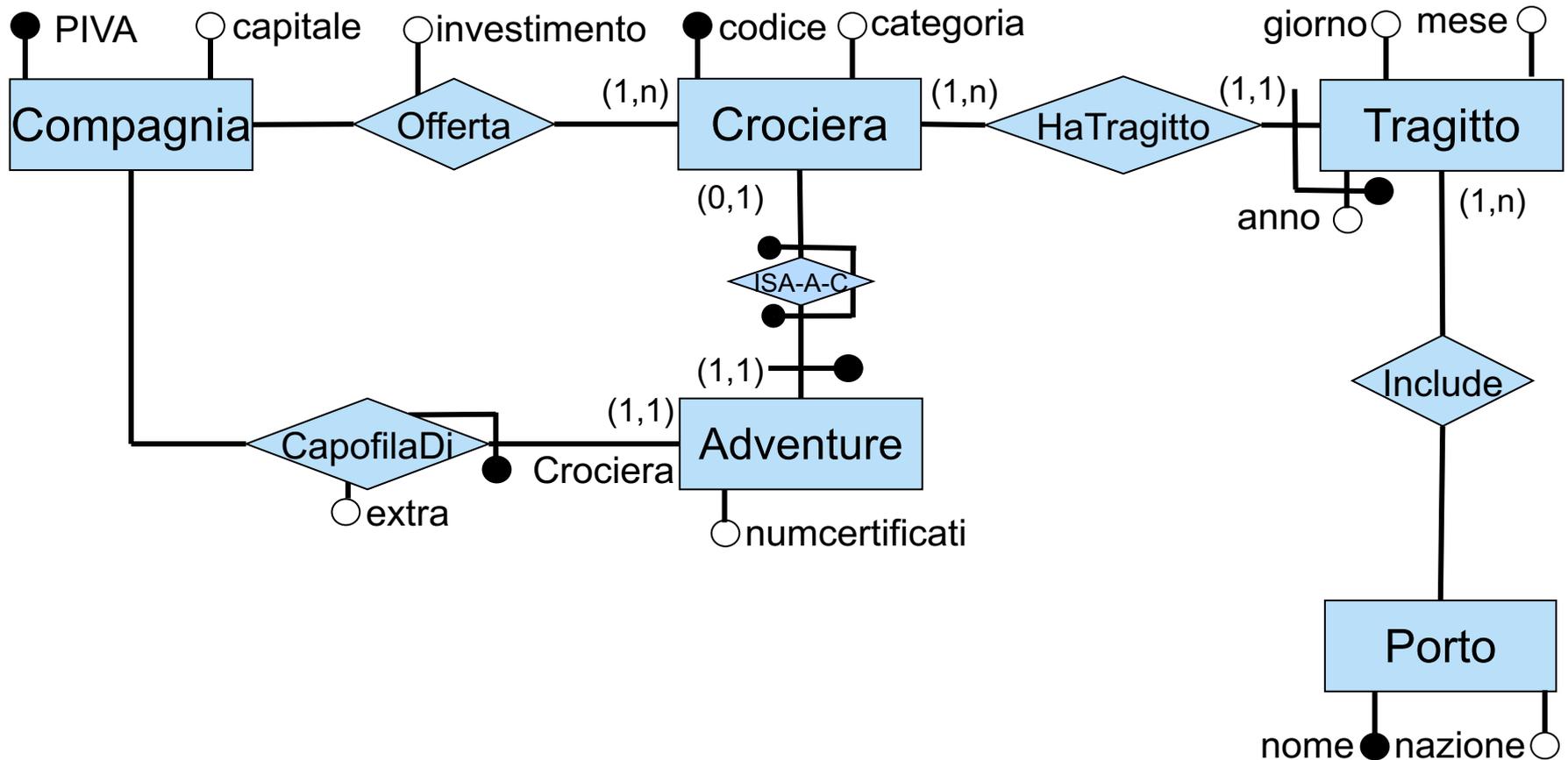
Schema concettuale:



**Nota:** nello schema della pagina precedente abbiamo assunto che anche alla prima determinazione del tragitto di una crociera sia associata una data. Nello schema concettuale mostrato in questa pagina abbiamo invece separato l'insieme dei porti del primo tragitto dagli insiemi di porti che emergono dai cambiamenti di tragitto. Anche la soluzione descritta in questa pagina è stata considerata corretta.

# Problema 2 – Ristrutturazione Schema ER

Schema concettuale ristrutturato:



**Vincolo esterno:** per ogni istanza  $\langle \text{Compagnia}:c, \text{Crociera}:a \rangle$  di CapofilaDi, esiste una istanza  $r$  di Crociera tale che  $\langle \text{Adventure}:a, \text{Crociera}:r \rangle$  è istanza di ISA-A-C e  $\langle \text{Compagnia}:c, \text{Crociera}:r \rangle$  è una istanza di Offerta.

# Problema 2 – Traduzione diretta

Schema logico prodotto dalla traduzione diretta:

Compagnia(piva, capitale)

Crociera(codice, categoria)

**inclusione:** Crociera[codice]  $\subseteq$  Offerta[crociera]

**inclusione:** Crociera[codice]  $\subseteq$  Tragitto[crociera]

Adventure(codice, numcertificati)

**foreign key:** Adventure[codice]  $\subseteq$  Crociera[codice]

**foreign key:** Adventure[codice]  $\subseteq$  Capofila[adventure]

Offerta(compagnia, crociera, investimento)

**foreign key:** Offerta[compagnia]  $\subseteq$  Compagnia[piva]

**foreign key:** Offerta[crociera]  $\subseteq$  Crociera[codice]

CapofilaDi(compagnia, crociera, extra)

**foreign key:** CapofilaDi[compagnia, crociera]  $\subseteq$  Offerta[compagnia, crociera]

**foreign key:** CapofilaDi[crociera]  $\subseteq$  Adventure[codice]

Tragitto(crociera, anno, giorno, mese)

**foreign key:** Tragitto[crociera]  $\subseteq$  Crociera[codice]

**inclusione:** Tragitto[crociera, anno]  $\subseteq$  Include[crociera, anno]

Porto(nome, nazione)

Include(crociera, anno, porto)

**foreign key:** Include[crociera, anno]  $\subseteq$  Tragitto[crociera, anno]

**foreign key:** Include[porta]  $\subseteq$  Porto[nome]

# Problema 2 – Ristrutturazione dello schema logico

## Schema logico prodotto dalla ristrutturazione:

1. La prima indicazione di progetto induce una decomposizione orizzontale di Crociera(codice,categoria) in CrocieraAdventure(codice,categoria) e CrocieraNonAdventure(codice,categoria).
2. La seconda indicazione induce un accorpamento tra la tabella CrocieraAdventure(codice,categoria) ottenuta al passo 1 e la tabella Adventure(codice,numcertificati) fortemente accoppiate, e tra la tabella così ottenuta (per la quale scegliamo il nome CrocieraAdventure) e CapofilaDi(compagnia,crociera,extra), anch'esse fortemente accoppiate.

Compagnia(piva,capitale)

CrocieraNonAdventure(codice,categoria)

**inclusione:** CrocieraNonAdventure[codice]  $\subseteq$  Offerta[crociera]

**inclusione:** CrocieraNonAdventure[codice]  $\subseteq$  Tragitto[crociera]

CrocieraAdventure(codice,categoria,numcertificati,capofila,extra)

**foreign key:** CrocieraAdventure[codice,capofila]  $\subseteq$  Offerta[crociera,compagnia]

**inclusione:** CrocieraAdventure[codice]  $\subseteq$  Tragitto[crociera]

**vincolo di disgiunzione:** CrocieraAdventure[codice]  $\cap$  CrocieraNonAdventure[codice] =  $\emptyset$

Offerta(compagnia,crociera,investimento)

**foreign key:** Offerta[compagnia]  $\subseteq$  Compagnia[piva]

**foreign key:** Offerta[crociera]  $\subseteq$  CrocieraAdventure[codice]  $\cup$  CrocieraNonAdventure[codice]

Tragitto(crociera,anno,giorno,mese)

**foreign key:** Tragitto[crociera]  $\subseteq$  CrocieraAdventure[codice]  $\cup$  CrocieraNonAdventure[codice]

**inclusione:** Tragitto[crociera,anno]  $\subseteq$  Include[crociera,anno]

Porto(nome,nazione)

Include(crociera,anno,porto)

**foreign key:** Include[crociera,anno]  $\subseteq$  Tragitto[crociera,anno]

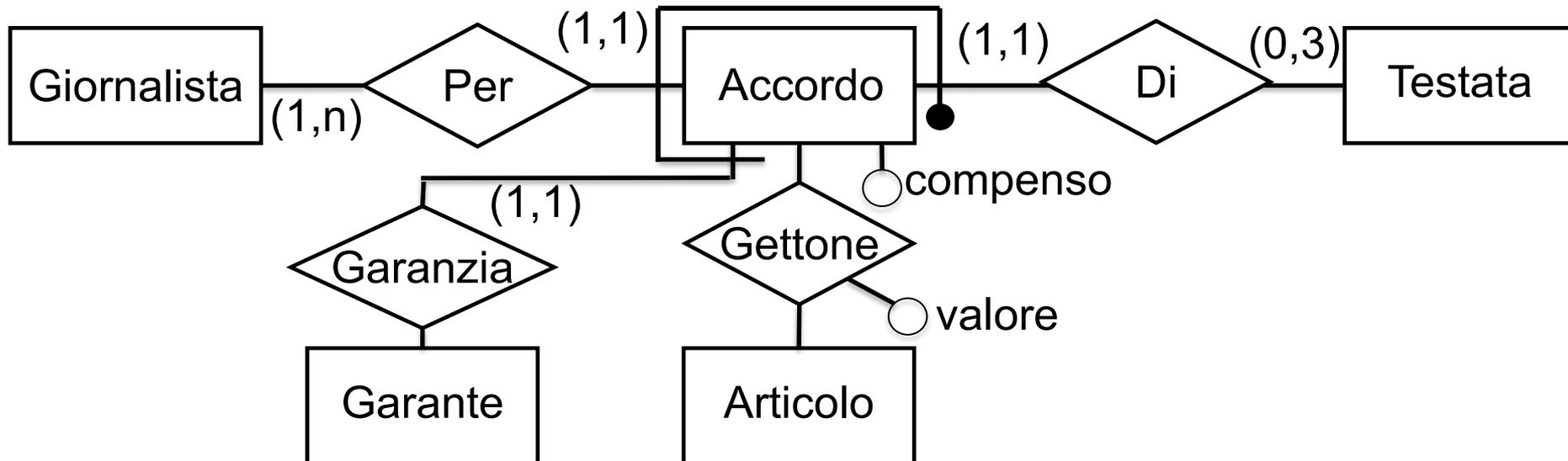
**foreign key:** Include[porto]  $\subseteq$  Porto[nome]

## Problema 3

Al fine di conformarsi ai nuovi requisiti occorre modellare una relazione tra gli accordi e gli articoli, in modo tale ad ogni coppia  $\langle c, t \rangle$  di questa nuova relazione possiamo associare il valore del gettone previsto dall'accordo  $c$  per l'articolo  $t$ . A partire dallo schema originario dobbiamo:

1. reificare Accordo, cioè trasformarla da una relazione in una entità;
2. introdurre l'entità Articolo e la relazione (che chiameremo Gettone) tra questa entità e la nuova entità Accordo, con un attributo per modellare il valore del gettone. In questo modo possiamo associare il valore del gettone ad ogni coppia  $\langle c, t \rangle$  nella relazione Gettone, dove  $c$  è un accordo e  $t$  è un articolo;
3. anche fissare a 3 la cardinalità massima dei legami di Testata con la nuova entità Accordo.

Lo schema risultante sarà:



# Problema 4 – testo e soluzione

Sia B una base di dati con le relazioni Corso(codice,disciplina) e PianoDiStudi(studente,codcorso).

(3.1) Scrivere in algebra relazionale una query su B che calcoli tutte le coppie (s,d) tali che nel piano di studi dello studente s non compare alcun corso della disciplina d.

(3.2) Scrivere in SQL una query su B che calcoli tutti gli studenti che hanno nel loro piano di studi almeno un corso di ogni disciplina.

Prodotto cartesiano

(3.1)  
 $\text{PROJ}_{\text{studente, disciplina}}(\text{PianoDiStudi JOIN Corso})$

-  
 $\text{PROJ}_{\text{studente, disciplina}}(\text{PianoDiStudi JOIN}_{\text{codcorso=codice}} \text{Corso})$

(3.2)  
select p.studente  
from PianoDiStudi p  
where not exists (select disciplina  
from Corso  
where disciplina not in (select disciplina  
from PianoDiStudi JOIN Corso  
on codcorso=codice  
where studente = p.studente)  
)

Disciplina che non compare nel piano di studi dello studente p.studente

## Problema 5

Per le definizioni di vincolo di integrità e di vincolo di integrità referenziale si rimanda alle slides del corso.

(5.1) Non esiste una base di dati corretta rispetto allo schema  $S$  tale che l'insieme  $\{B,C\}$  non soddisfa la condizione di superchiave in  $R$ . Dimostriamo questa proprietà facendo vedere che se in una base di dati  $T$  l'insieme  $\{B,C\}$  non soddisfa la condizione di superchiave in  $R$ , allora  $T$  non è corretta rispetto ad  $S$ . Infatti, una base di dati  $T$  in cui l'insieme  $\{B,C\}$  non soddisfa la condizione di superchiave in  $R$  contiene necessariamente due tuple  $(a_1,b,c)$  e  $(a_2,b,c)$  in  $R$ , con  $a_1 \neq a_2$ . Ma a questo punto, o queste due tuple non appartengono a  $Q$  e allora  $T$  non soddisfa il vincolo di foreign key e  $T$  non è corretta rispetto ad  $S$ , oppure queste due tuple appaiono anche in  $Q$  e quindi in  $T$  la relazione  $Q$  contiene due tuple  $(a_1,b,c)$  e  $(a_2,b,c)$ , con  $a_1 \neq a_2$ , il vincolo di chiave primaria in  $Q$  è violato e anche in questo caso  $T$  non è corretta rispetto ad  $S$ . Abbiamo, quindi, dimostrato che non può esistere una base di dati corretta rispetto ad  $S$  in cui l'insieme  $\{B,C\}$  non soddisfa la condizione di superchiave in  $R$ .

(5.2) È facile verificare che esiste una base di dati  $T$  corretta rispetto ad  $S$  in cui l'insieme  $\{D\}$  non soddisfa la condizione di superchiave. Definiamo semplicemente  $T$  come la base di dati in cui  $R$  è vuota e  $Q$  contiene le due tuple  $(d,e_1,f)$  e  $(d,e_2,f)$  con  $e_1 \neq e_2$ . È immediato verificare che  $T$  soddisfa tutti i vincoli di  $S$  ed è quindi corretta rispetto allo schema  $S$  ed in essa l'insieme  $\{D\}$  non soddisfa la condizione di superchiave.

(5.3) Non esiste una base di dati corretta rispetto allo schema  $S$  tale che l'insieme  $\{F\}$  non soddisfa la condizione di superchiave in  $R$ . Dimostriamo questa proprietà facendo vedere che se in una base di dati  $T$  l'insieme  $\{F\}$  non soddisfa la condizione di superchiave in  $Q$ , allora  $T$  non è corretta rispetto ad  $S$ . Infatti, una base di dati  $T$  in cui l'insieme  $\{F\}$  non soddisfa la condizione di superchiave in  $Q$  contiene necessariamente due diverse tuple con lo stesso valore per  $F$ , ossia due tuple diverse del tipo  $(d_1,e_1,f)$  e  $(d_2,e_2,f)$ . Ma a questo punto, o queste due tuple non soddisfano il vincolo di tupla definito in  $Q$  e allora  $T$  non è corretta rispetto ad  $S$ , oppure queste due tuple soddisfano il vincolo di tupla definito in  $Q$  e allora  $e_1 = e_2 = f$  e siccome sono due tuple diverse abbiamo che  $d_1 \neq d_2$ , il che implica l'esistenza di due tuple diverse in  $Q$  con lo stesso valore di  $E$  e quindi la violazione del vincolo di chiave primaria in  $Q$ . Abbiamo, quindi, dimostrato che non esiste una base di dati corretta rispetto ad  $S$  tale che l'insieme  $\{F\}$  non soddisfa la condizione di superchiave in  $Q$ .