Corso di Basi di dati

Secondo Compitino - Fila A - Soluzione

6 giugno 2008

1. Comune(codC,*nomeC*, *provincia*, regione, numAb)
   Risiede(codF, cognome, nome, dataN, via, nCiv, nInt, codC$^{\text{Comune}}$)
   Fiume(nomeF, lunghezza, sorgente$^{\text{Comune}}$, foce$^{\text{Comune}}$)
   Bagna(nomeF$^{\text{Fiume}}$, codC$^{\text{Comune}}$)

   Nomi dei comuni della provincia di Pavia con meno di 1000 abitanti bagnati da un fiume lungo meno di 200 km in cui abita almeno una persona che ha il nome uguale al cognome (es. Armando Armando)

   $$\{t : \{nomeC\} \mid (\exists c)((\exists r)((\exists f)((\exists b)(c \in Comune \wedge r \in Risiede \wedge f \in Fiume \wedge b \in Bagna \wedge c[provincia] =' pv' \wedge c[numAb] < 1000 c[codC] = r[codC] \wedge r[nome] = r[cognome] \wedge c[codC] = b[codC] \wedge b[nomeF] = f[nomeF] \wedge f[lunghezza] < 200))))\}$$

2. (a)
```
CREATE TABLE Comune
(codC      NUMERIC(4,0) PRIMARY KEY,
 nomeC     VARCHAR(30),
 provincia VARCHAR(20),
 regione   VARCHAR(20),
 numAb     NUMERIC(7,0),
 UNIQUE (nomeC,provincia));

CREATE TABLE Fiume
(nomeF     VARCHAR(30) PRIMARY KEY,
 lunghezza NUMERIC(8,2),
 sorgente  NUMERIC(4,0) REFERENCES Comune,
 foce  NUMERIC(4,0) REFERENCES Comune
 CHECK (sorgente <> foce));
```

   (b) i. il numero degli abitanti di un comune deve coincidere con il numero delle persone che vi hanno la residenza

```
CREATE ASSERTION numAbOK
CHECK (NOT EXISTS (SELECT *
                   FROM Comune C
                   WHERE numAb <> (SELECT COUNT(*)
                                   FROM Risiede
                                   WHERE codC = C.codC));
```

   ii. un fiume non puó sorgere e sfociare nello stesso comune
   vedi vincolo `CHECK (sorgente <> foce)` nella tabella `Fiume`

   (c) i. determinare i nomi delle regioni in cui non sorge alcun fiume lungo piú di 200 km

```
SELECT regione
FROM Comune
WHERE regione NOT IN (SELECT regione
                      FROM Comune JOIN Fiume ON codC = sorgente
                      WHERE lunghezza > 200);
```

   ii. determinare i fiumi che bagnano almeno quattro regioni diverse, insieme al numero totale di abitanti dei comuni bagnati da quel fiume

```
SELECT nomeF, SUM(numAb)
FROM Bagna NATURAL JOIN Comune
GROUP BY nomeF
HAVING COUNT(DISTINCT regione) >= 4;
```

   iii. determinare il comune piú popoloso tra i comuni piemontesi bagnati da almeno due fiumi

```
SELECT codC
FROM Comune
WHERE regione = 'piemonte'
```

```
AND codC IN (SELECT codC
             FROM Bagna
             GROUP BY codC
             HAVING COUNT(*) >= 2)
AND numAb >= ALL(SELECT numAb
                 FROM Comune
                 WHERE regione = 'piemonte'
                       AND codC IN (SELECT codC
                                    FROM Bagna
                                    GROUP BY codC
                                    HAVING COUNT(*) >= 2);
```

iv. determinare per ogni regione, la provincia di tale regione bagnata dal maggior numero di fiumi

```
SELECT regione, provincia
FROM Comune C NATURAL JOIN Bagna
GROUP BY provincia, regione
HAVING COUNT(DISTINCT nomeF) >= ALL (SELECT COUNT(DISTINCT nomeF)
                                     FROM Comune NATURAL JOIN Bagna
                                     WHERE regione = C.regione
                                     GROUP BY provincia);
```

3. RIPARAZIONE(numS, marca, modello, codFProp, nomeProp, telProp, guasto, dataRip, importoRip, tecnico)

(a) i. apparecchiature di diverse marche e modelli possono essere soggette allo stesso tipo di guasto
non formalizzabile tramite dipendenza funzionale

ii. un'apparecchiatura puó subire riparazioni per guasti differenti nella stessa data
non formalizzabile tramite dipendenza funzionale

iii. ogni guasto di ogni apparecchiatura viene riparato al piú una volta per ogni data
guasto numS dataRip → importoRip tecnico

iv. gli apparecchi di una stessa marca e modello sono riparati sempre dallo stesso tecnico
marca modello → tecnico

v. l'importo della riparazione dipende dal tipo di guasto, dalla marca e modello dell'apparecchio e dalla data della riparazione
guasto marca modello dataRip → importoRip

vi. ogni apparecchiatura (identificata da numero di serie) ha una certa marca, modello e proprietario
numS → marca modello codFProp

vii. ogni proprietario (identificato da codice fiscale) ha un nome e numero di telefono
codFProp → nomeProp telProp

viii. ogni numero di telefono corrisponde ad un unico proprietario
telProp → codFProp

(b) insieme minimale:
marca modello → tecnico
guasto marca modello dataRip → importoRip
numS → marca
numS → modello
numS → codFProp
codFProp → nomeProp
codFProp → telProp
telProp → codFProp

(c) chiave: (numS, guasto, dataRip)

(d) lo schema non é in 3NF né in BCNF

(e) scomposizione lossless join dello schema in BCNF:
(codFProp, nomeProp, telProp), (numS, marca, modello, codFProp), (numS, guasto, dataRip, importoRip, tecnico)
non preserva ad es. la dipendenza marca modello → tecnico

(f) scomposizione lossless join dello schema in 3NF che preserva le dipendenze
(codFProp, nomeProp, telProp), (numS, marca, modello, codFProp), (marca, modello, tecnico), (guasto, marca, modello, dataRip, importoRip), (numS, guasto, dataRip)

4. $R = (A, B, C, D, E, F)$, dipendenze funzionali:

$$B\ C \to F \qquad B\ D \to A \qquad B\ E \to C \qquad C \to D \qquad D \to E$$

(a) no, controesempio:

$R =$

A	B	C	D	E	F
a_1	b_1	c_1	d_1	e_1	f_1
a_2	b_2	c_1	d_1	e_1	f_2

$\Pi_{ABC}(R) \bowtie \Pi_{CDEF}(R) =$

A	B	C	D	E	F
a_1	b_1	c_1	d_1	e_1	f_1
a_2	b_2	c_1	d_1	e_1	f_2
a_1	b_1	c_1	d_1	e_1	f_2
a_2	b_2	c_1	d_1	e_1	f_1

(b) l'insieme di dipendenze è in forma minimale

(c) chiavi: BC, BD, BE

(d) lo schema é in 3NF ma non in BCNF

(e) scomposizione lossless join in BCNF: (DE), (CD), $(ABCF)$ non preserva ad es. la dipendenza $BD \to C$

(f) lo schema è già in 3NF