Sistemi Informativi T
21 febbraio 2012
Risoluzione

Tempo a disposizione: 2:30 ore

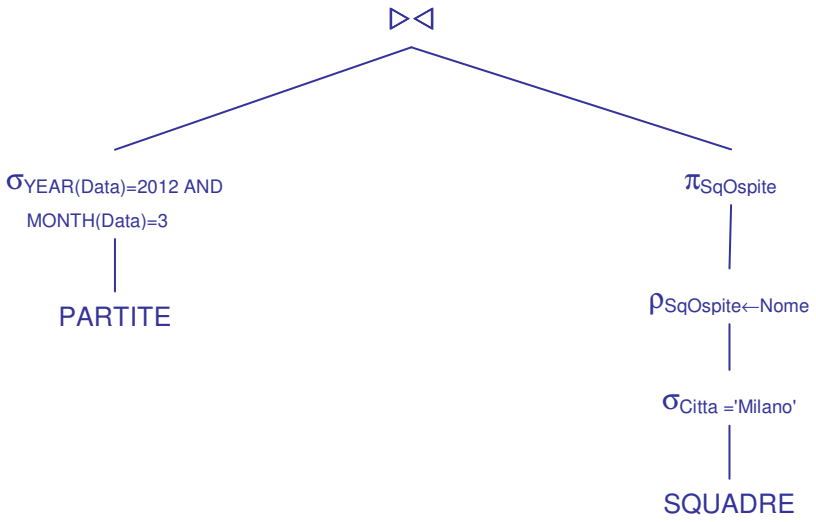
1) Algebra relazionale (3 punti totali):

Date le seguenti relazioni:

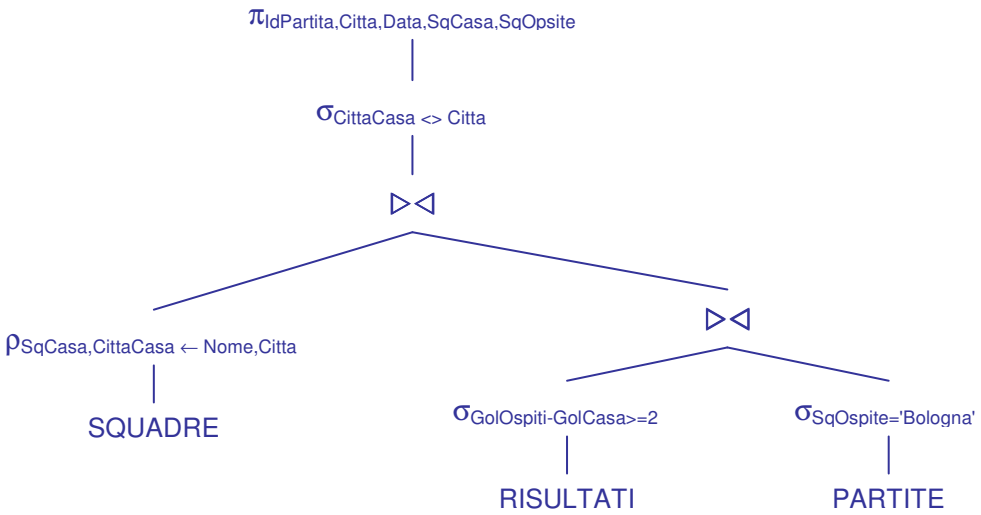
```
SQUADRE(Nome,Citta);
PARTITE(IdPartita,Citta,Data,SqCasa,SqOspite),
SqCasa REFERENCES SQUADRE, SqOspite REFERENCES SQUADRE;
RISULTATI(IdPartita,GolCasa,GolOspiti);
IdPartita REFERENCES PARTITE
```

si scrivano in algebra relazionale le seguenti interrogazioni:

1.1) [1 p.] Le partite che le squadre di Milano giocano fuori casa a Marzo 2012



1.2) [2 p.] Le partite in cui la squadra del Bologna è stata ospite e ha vinto con almeno 2 gol di scarto, giocate in campo neutro (città diversa da quella della squadra ospitante)



Si noti la ridenominazione di entrambi gli attributi di SQUADRE, necessaria per eseguire correttamente il join.

Sistemi Informativi T
21 febbraio 2012
Risoluzione

2) SQL (5 punti totali)

Con riferimento al DB dell'esercizio 1, si scrivano in SQL le seguenti interrogazioni:

2.1) [2 p.] Il numero complessivo di gol segnati in trasferta, per ogni squadra che ne ha segnati in trasferta più del Bologna

```
SELECT   SqOspite, SUM(GolOspiti) AS GolTrasferta
FROM     PARTITE P, RISULTATI R
WHERE    P.IdPartita = R.IdPartita
GROUP BY SqOspite
HAVING   SUM(GolOspiti) > ( SELECT SUM(GolOspiti)
                            FROM   PARTITE P1, RISULTATI R1
                            WHERE  P1.IdPartita = R1.IdPartita
                            AND    P1.SqOspite = 'Bologna' )
```

2.2) [3 p.] Considerando il numero complessivo di gol segnati in una data (GOLDATA), si riporti quante volte ogni valore di GOLDATA si è verificato

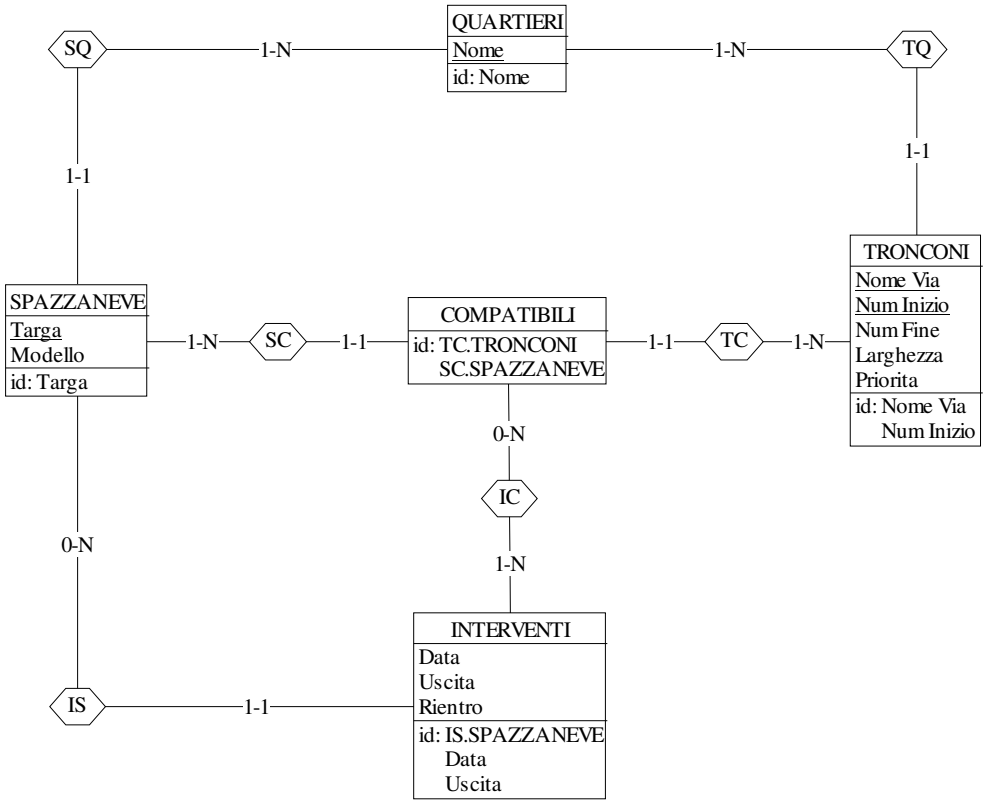
```
WITH GOLPERDATA (DATA, GOLDATA) AS (
   SELECT P.Data, SUM(R.GolCasa + R.GolOspiti)
   FROM   PARTITE P, RISULTATI R
   WHERE  P.IdPartita = R.IdPartita
   GROUP BY  P.Data )
SELECT GOLDATA, COUNT(*) AS NUMVOLTE
FROM   GOLPERDATA
GROUP BY GOLDATA

-- Nella Common Table Expression si contano i gol di ogni giornata
```

Sistemi Informativi T
21 febbraio 2012
Risoluzione

3) Progettazione concettuale (6 punti)

La cooperativa di Spazzaneve StradePuliteAdesso! (SPA) è organizzata in modo da garantire sempre interventi tempestivi nella sua città. Ad ogni quartiere sono associati diversi mezzi spazzaneve, e tutti i quartieri sono serviti. In funzione delle specifiche condizioni, i mezzi possono servire diverse strade del quartiere, o parte di esse nel caso di strade molto lunghe (queste strade, ai fini del servizio, sono frazionate in "tronconi" identificati dai numeri civici di inizio e fine; ad es. Via Emilia dal 35 al 216). Ogni strada ha una "priorità" fissa che serve a stabilire quali prima servire. Informazioni sulla larghezza della carreggiata servono a stabilire quali mezzi possono transitare sulle diverse strade. Per migliorare il suo servizio, la SPA mantiene, per ogni giorno in cui è intervenuta, traccia dei mezzi impiegati. Per ogni intervento di un mezzo (anche più di uno in un giorno) si tiene traccia dei tempi di uscita e di rientro e delle strade effettivamente servite.

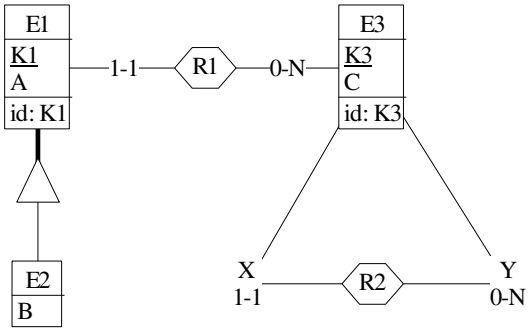


Commenti:

- Lo schema proposto distingue una parte "statica" (in alto nel disegno) e una "dinamica", rappresentata dall'entità INTERVENTI.
- Vale il vincolo (non rappresentabile nello schema) che uno spazzaneve può intervenire solo su tronconi con esso compatibili. In altri termini, IS.SPAZZANEVE = SC.SPAZZANEVE per le istanze associate tramite IC.
- In una soluzione alternativa, ogni istanza "statica" di COMPATIBILI avrebbe potuto dar luogo a N istanze ("dinamiche") di INTERVENTI, se per quest'ultima si fosse inserito nell'identificatore anche il troncone (e cardinalità 1-1 verso IC). Ma in questo caso ogni istanza di INTERVENTI avrebbe modellato l'intervento su un singolo troncone, e quindi, per rappresentare le informazioni di Uscita e Rientro sarebbe stato necessario introdurre un'altra entità.

Sistemi Informativi T
21 febbraio 2012
Risoluzione

4) Progettazione logica (6 punti totali)



Dato lo schema concettuale in figura e considerando che:

a) tutti gli attributi sono di tipo INT;
b) le associazioni R1 e R2 non vengono tradotte separatamente;
c) le entità E1 ed E2 vengono tradotte assieme;
d) un'istanza di E1 non è mai associata, tramite R1, a un'istanza di E3 che partecipa nell'associazione R2 con il ruolo Y;

4.1) [3 p.] Si progettino gli opportuni schemi relazionali e si definiscano tali schemi in DB2 (sul database SIT_STUD) mediante un file di script denominato SCHEMI.txt

```
CREATE TABLE E3 (
K3 INT NOT NULL PRIMARY KEY,
C INT NOT NULL,
K3Y INT NOT NULL REFERENCES E3          );  -- foreign key verso l'istanza che partecipa nel ruolo Y

CREATE TABLE E1 (
K1 INT NOT NULL PRIMARY KEY,
A INT NOT NULL,
K3 INT NOT NULL REFERENCES E3,
TIPO2 SMALLINT NOT NULL CHECK (TIPO2 IN (0,1)),          -- 1: istanza anche di E2
B INT,
CONSTRAINT E2 CHECK
     ( (TIPO2 = 1 AND B IS NOT NULL) OR  (TIPO2 = 0 AND B IS NULL) )        );
```

4.2) [3 p.] Per i vincoli non esprimibili a livello di schema si predispongano opportuni **trigger che evitino inserimenti di tuple non corrette**, definiti in un file TRIGGER.txt e usando il simbolo '@' per terminare gli statement SQL

```
-- Per garantire il rispetto del vincolo di cui al punto d) è necessario impostare il seguente trigger:
CREATE TRIGGER PUNTO_D
NO CASCADE BEFORE INSERT ON E1
REFERENCING NEW AS N
FOR EACH ROW
WHEN (EXISTS ( SELECT *
                    FROM E3
                   WHERE E3.K3Y = N.K3) )
SIGNAL SQLSTATE '70001' ('La tupla non deve referenziare una istanza di E3 che partecipa in R2 come Y!')@
```