Sistemi Informativi T
3 luglio 2012
Risoluzione

Tempo a disposizione: 2:30 ore

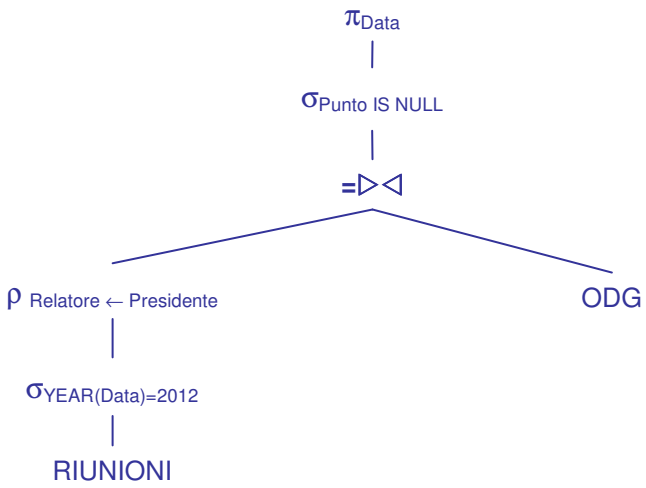
1) Algebra relazionale (3 punti totali):

Date le seguenti relazioni:

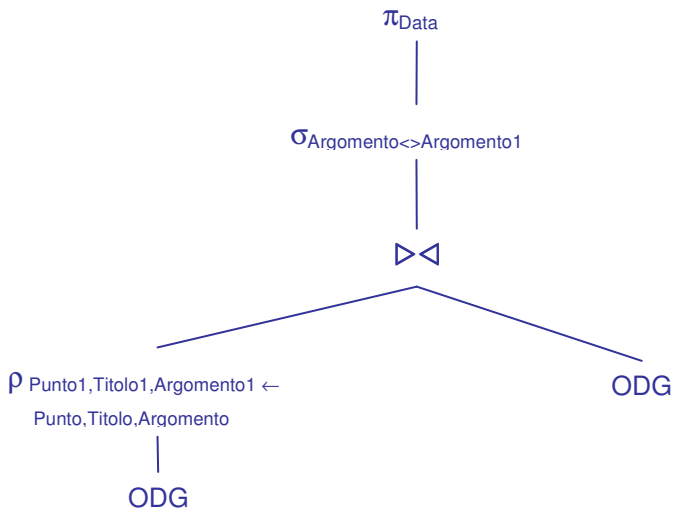
```
RIUNIONI(Data,Presidente);
PRESENZE(Data,Persona),
Data REFERENCES RIUNIONI;
ODG(Data,Punto,Titolo,Argomento,Relatore),
Data,Relatore REFERENCES PRESENZE;
-- ODG = Ordine Del Giorno
-- Presidente, Persona e Relatore sono definiti sullo stesso dominio
```

si scrivano in algebra relazionale le seguenti interrogazioni:

1.1) [1 p.] Le riunioni del 2012 in cui il presidente non ha relazionato su nessun punto



1.2) [2 p.] Le riunioni in cui una persona ha relazionato su almeno due diversi argomenti



Sistemi Informativi T
3 luglio 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.] Per ogni persona, gli argomenti su cui non ha mai relazionato, supponendo che su ogni argomento ci sia stata almeno una relazione

```
SELECT   DISTINCT P.Persona, O.Argomento
FROM     PRESENZE P, ODG O
WHERE    NOT EXISTS ( SELECT *
                      FROM   ODG O1
                      WHERE  O1.Argomento = O.Argomento
                      AND    O1.Relatore  = P.Persona    )
-- La soluzione consiste nel rimuovere dal prodotto Cartesiano generato
-- nel blocco esterno le coppie (Persona,Argomento) che si ritrovano in ODG
```

2.2) [3 p.] Per ogni argomento, il relatore che lo ha trattato il maggior numero di volte, escludendo dal conteggio le volte che il relatore era anche presidente della riunione

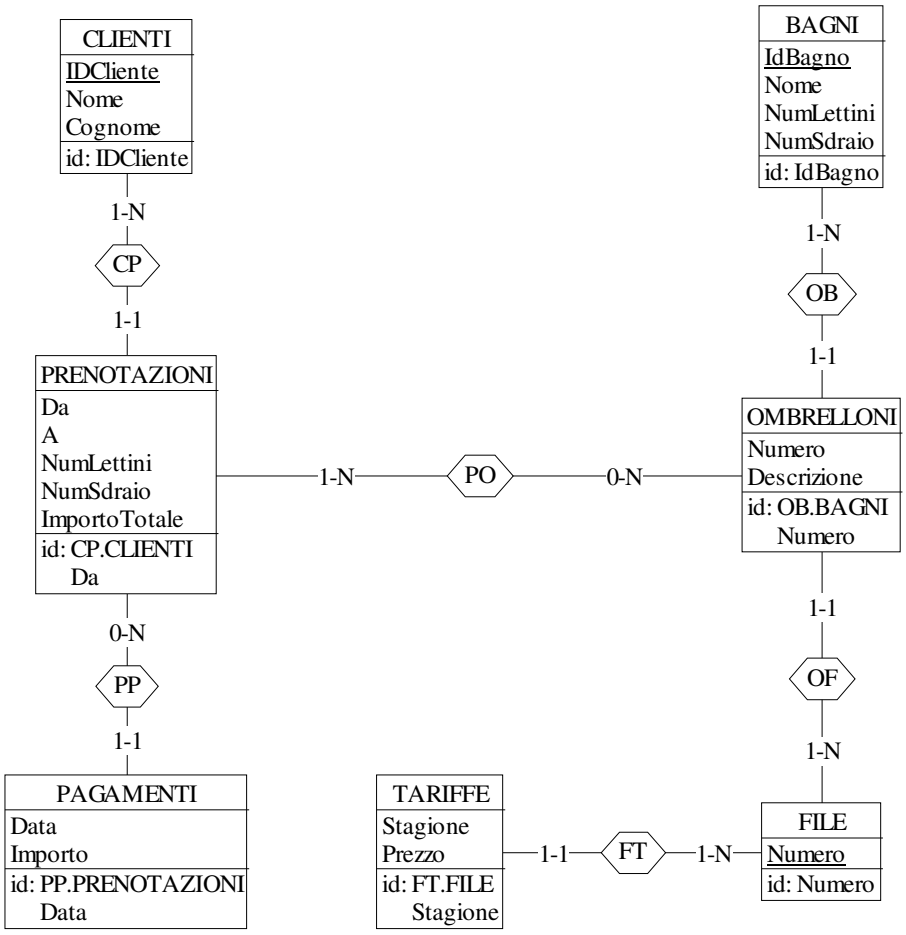
```
WITH ARGREL (Relatore, Argomento, NumVolte) AS (
   SELECT O.Relatore,O.Argomento, COUNT(*)
   FROM   ODG O, RIUNIONI R
   WHERE  O.Relatore <> R.Presidente
   AND    O.Data = R.Data
   GROUP BY  O.Relatore,O.Argomento )
SELECT A.Argomento, A.Relatore
FROM   ARGREL A
WHERE  A.NumVolte >= ALL ( SELECT A1.NumVolte
                           FROM   ARGREL A1
                           WHERE  A1.Argomento = A.Argomento )

-- Nella Common Table Expression si contano le volte in cui un dato relatore
-- ha relazionato su un dato argomento quando non era presidente
```

Sistemi Informativi T
3 luglio 2012
Risoluzione

3) Progettazione concettuale (6 punti)

La cooperativa di bagnini FaReSol gestisce una serie di stabilimenti balneari (o "bagni"). In ogni periodo di tempo, un cliente può prenotare uno o più ombrelloni (ogni ombrellone ha un numero univoco in un bagno), e anche lettini e sedie sdraio (questi non hanno nulla che li identifichi, in quanto per ogni bagno è noto solo il loro numero complessivo). Le tariffe giornaliere per il noleggio degli ombrelloni variano in funzione della stagione e della fila dell'ombrellone, e sono le stesse per tutti i bagni della FaReSol. Il pagamento degli oggetti affittati può avvenire in diverse soluzioni (ad es. acconto del 10% all'atto della prenotazione, 30% a metà vacanza, e il saldo al termine del periodo di soggiorno), ed è quindi necessario tener traccia dei pagamenti effettuati.



Commenti:

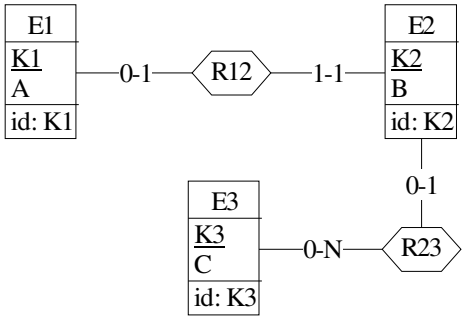
- La soluzione proposta utilizza numerose identificazioni esterne, al fine di modellare nella maniera più precisa le specifiche senza fare ricorso a codici identificativi non strettamente necessari (ad es. per PRENOTAZIONI)
- Non è rappresentabile il vincolo (implicito) che gli ombrelloni prenotati in uno stesso periodo da uno stesso cliente facciano tutti parte dello stesso bagno. Si noti che se tale vincolo non venisse considerato, allora gli attributi NumLettini e NumSdraio non potrebbero essere parte dell'entità PRENOTAZIONI, ma dovrebbero essere inseriti come attributi dell'associazione PO (altrimenti non si potrebbe sapere quanti lettini e sdraio vengono prenotati nei diversi bagni)

Sistemi Informativi T
3 luglio 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) l'associazione R23 non viene tradotta separatamente;
c) le entità E1 ed E2 vengono tradotte assieme;
d) per le solo istanze di E2 che partecipano a R23, esiste una dipendenza funzionale da B a C;



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     );

CREATE TABLE E12 (
K1 INT NOT NULL PRIMARY KEY,
A INT NOT NULL,
TIPO2 SMALLINT NOT NULL CHECK (TIPO2 IN (0,1)),          -- 1: istanza anche di E2
K2 INT,
B INT,
K3 INT REFERENCES E3,
CONSTRAINT E2 CHECK
     ( (TIPO2 = 1 AND K2 IS NOT NULL AND B IS NOT NULL) OR
      (TIPO2 = 0 AND K2 IS NULL AND B IS NULL AND K3 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

```
-- Unicita' dei valori di K2 (chiave con valori nulli):
CREATE TRIGGER K2_KEY
NO CASCADE BEFORE INSERT ON E12
REFERENCING NEW AS N
FOR EACH ROW
WHEN (EXISTS (     SELECT *
                   FROM E12
                   WHERE E12.K2 = N.K2) )
SIGNAL SQLSTATE '70001' ('K2 non ammette valori duplicati!')@

-- Per garantire il rispetto del vincolo di cui al punto d) è necessario impostare il seguente trigger:
CREATE TRIGGER FD_B_C
NO CASCADE BEFORE INSERT ON E12
REFERENCING NEW AS N
FOR EACH ROW
WHEN (EXISTS (     SELECT *
                   FROM E12, E3 E3X, E3 E3Y
                   WHERE E12.K3 = E3X.K3
                   AND N.K3 = E3Y.K3
                   AND E12.B = N.B
                   AND E3X.C <> E3Y.C) )
SIGNAL SQLSTATE '70002' ('La tupla non rispetta la FD da B a C!')@
-- Si noti che E3 viene usata 2 volte, una per trovare il valore di C (E3X.C) associato a un dato valore di B,
-- l'altra per trovare il valore di C (E3Y.C) associato alla nuova tupla inserita in E12
```