Sistemi Informativi T
13 giugno 2011
Risoluzione

Tempo a disposizione: 2:30 ore

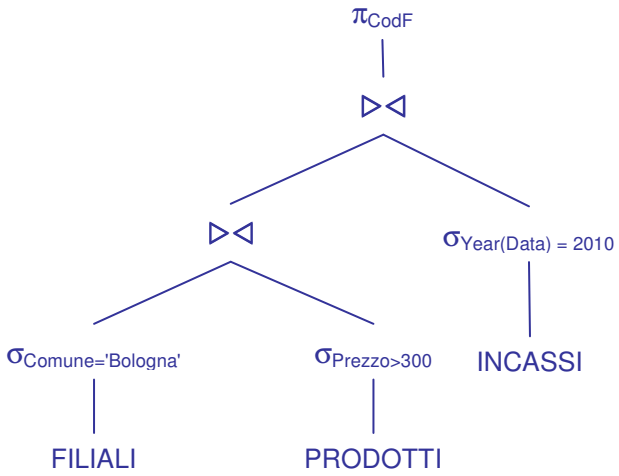
1) Algebra relazionale (3 punti totali):

Date le seguenti relazioni:

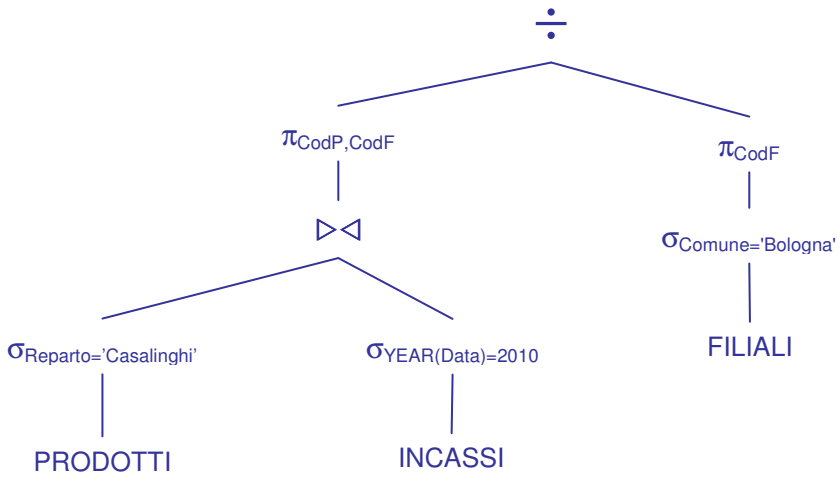
```
FILIALI(CodF,Indirizzo,Comune);
PRODOTTI(CodP,Reparto,Prezzo);
INCASSI(CodF,CodP,Data,Importo),
CodF REFERENCES FILIALI, CodP REFERENCES PRODOTTI;
-- Importo e Prezzo sono interi > 0
```

si scrivano in algebra relazionale le seguenti interrogazioni:

1.1) **[1 p.]** Tutte le filiali di Bologna che hanno venduto almeno un prodotto con un prezzo maggiore di 300 € nel 2010



1.2) **[2 p.]** I codici dei prodotti del reparto Casalinghi che sono stati venduti nel 2010 almeno una volta da tutte le filiali di Bologna



Sistemi Informativi T
13 giugno 2011
Risoluzione

2) **SQL (5 punti totali)**

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

2.1) [2 p.] I codici dei prodotti del reparto Casalinghi che sono stati venduti nel 2010 almeno una volta da tutte le filiali di Bologna

```
SELECT   P.CodP
FROM     PRODOTTI P, INCASSI I, FILIALI F
WHERE    P.CodP = I.CodP
AND      I.CodF = F.CodF
AND      P.Reparto = 'Casalinghi'
AND      F.Comune = 'Bologna'
AND      YEAR(I.Data) = 2010
GROUP BY P.CodP
HAVING   COUNT(DISTINCT F.CodF) = ( SELECT COUNT(*)
                                    FROM FILIALI F1
                                    WHERE F1.Comune = 'Bologna' )
```

2.2) [3 p.] Per ogni giorno del 2010 ed ogni filiale, il reparto che ha dato luogo al maggior importo di vendite

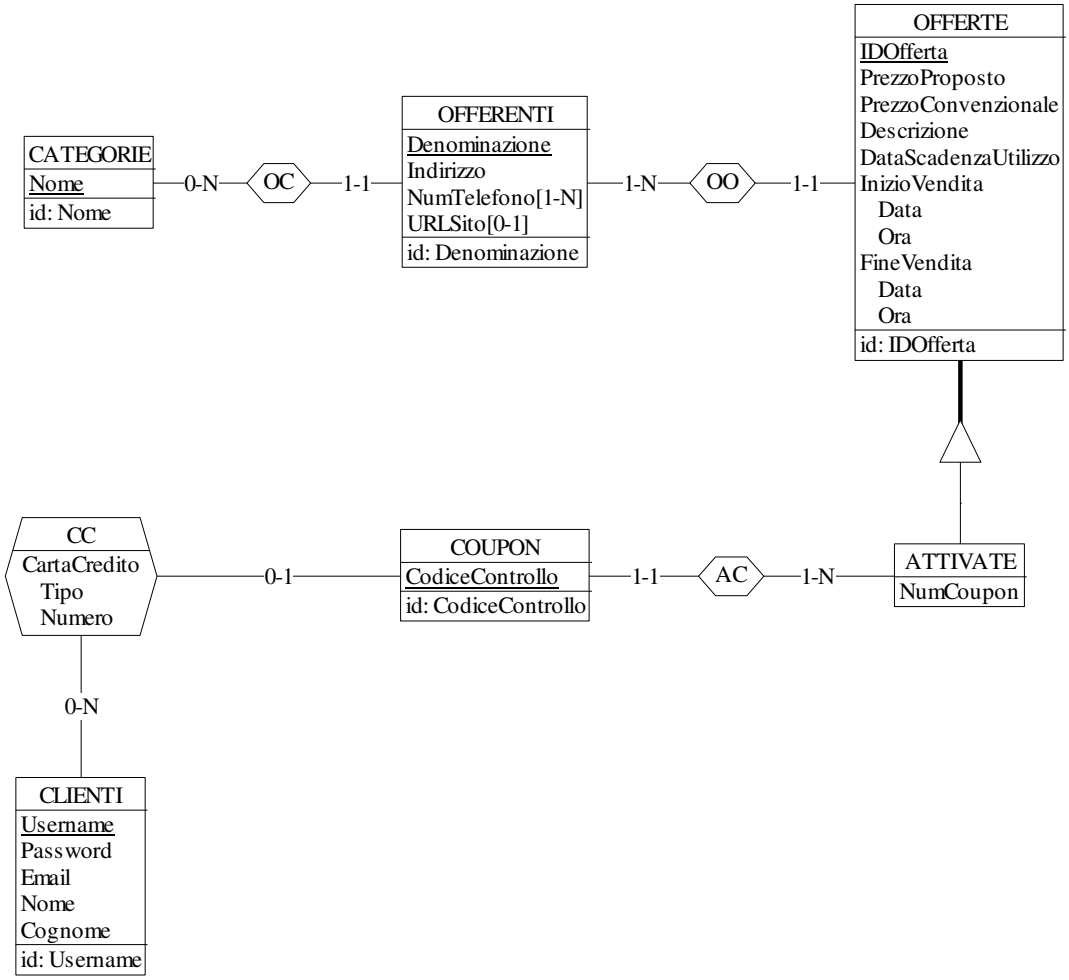
```
WITH
TOTIMPORTI(CodF,Giorno,Reparto,TotImporto) AS (
   SELECT I.CodF, I.Data, P.Reparto, SUM(I.Importo)
   FROM   PRODOTTI P, INCASSI I
   WHERE  P.CodP = I.CodP
   AND    YEAR(I.Data) = 2010
   GROUP BY  I.CodF, I.Data, P.Reparto )
SELECT T.CodF,T.Giorno,T.Reparto
FROM   TOTIMPORTI T
WHERE  T.TotImporto = ( SELECT MAX(T1.TotImporto)
                        FROM TOTIMPORTI T1
                        WHERE T1.CodF = T.CodF
                        AND T1.Giorno = T.Giorno )

-- La Common Table Expression calcola, per ogni giorno del 2010, ogni
-- filiale e ogni reparto, l'importo totale delle relative vendite.
-- Il blocco SELECT confronta quindi le vendite totali a parita' di
-- giorno e filiale
```

Sistemi Informativi T
13 giugno 2011
Risoluzione

3) Progettazione concettuale (6 punti)

Il sito OfferteOnLine (OOL) mette a disposizione dei propri clienti (regsitrati con username, password, email, nome e cognome) una serie di offerte di servizi a condizioni particolarmente vantaggiose. Ogni offerta fa capo a un offerente di una determinata categoria commerciale (ristoranti, alberghi, centri estetici, ecc.) e si caratterizza per il prezzo proposto, il prezzo convenzionale, la descrizione dell'offerta e la data entro cui si può utilizzare l'offerta (ad es.: cena per 2 persone a 35€ anziché 60€ entro il 31/07/2011). Ogni offerente mette a disposizione sul sito OOL le necessarie informazioni di contatto (indirizzo, uno o più numeri di telefono e eventualmente l'URL del proprio sito). Un cliente che acquista (necessariamente con carta di credito) un'offerta riceve il relativo coupon via email, con i dettagli dell'offerta e un codice di controllo, specifico del coupon. I codici sono stabiliti all'atto dell'attivazione dell'offerta sul sito OOL. Per ogni offerta, che ovviamente ha una data e ora di attivazione e di fine, c'è un numero massimo di coupon vendibili.
Per favorire la visibilità delle offerte, queste possono essere pubblicizzate sul sito OOL come offerte future, ossia prima della data di attivazione. In questo caso i codici di controllo non sono ancora registrati sul sito.

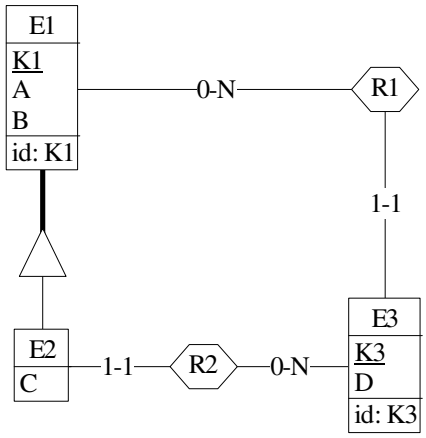


Commenti:

- L'esercizio non presenta difficoltà di rilievo
- Si è assunta la possibilità di pagare ogni COUPON con una carta di credito specifica
- Si è inoltre assunto che anche per le offerte future siano definite ora e data di inizio e fine vendita, che quindi compaiono come attributi comuni nell'entità OFFERTE. Ciò rende inutile l'introduzione di una sotto-entità specifica per le offerte future

Sistemi Informativi T
13 giugno 2011
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 E3 non è mai associata, tramite R1, a un'istanza di E1 con A > 10;
e) vale la dipendenza funzionale A → B, ma lo schema non viene normalizzato;

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 E1(
K1 INT NOT NULL PRIMARY KEY,
A INT NOT NULL,
B INT NOT NULL,
TIPO SMALLINT NOT NULL CHECK (TIPO IN (1,2)),          -- TIPO = 2: istanza anche di E2
C INT,
K3 INT,                                  -- il vincolo di foreign key va aggiunto dopo aver definito E3
CONSTRAINT E2 CHECK (
     (TIPO = 1 AND C IS NULL AND K3 IS NULL) OR
     (TIPO = 2 AND C IS NOT NULL AND K3 IS NOT NULL))          );

CREATE TABLE E3(
K3 INT NOT NULL PRIMARY KEY,
D INT NOT NULL,
K1 INT NOT NULL REFERENCES E1          );

ALTER TABLE E1
ADD CONSTRAINT FKR2 FOREIGN KEY (K3) REFERENCES E3 ;
```

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

```
CREATE TRIGGER INSERT_E3        -- vincolo al punto d)
NO CASCADE BEFORE INSERT ON E3
REFERENCING NEW AS N
FOR EACH ROW MODE DB2SQL
WHEN (EXISTS (SELECT *
                    FROM E1
                    WHERE E1.K1 = N.K1
                    AND E1.A > 10 ))
SIGNAL SQLSTATE '70001' ('L'istanza associata di E1 ha valore A > 10!')@

CREATE TRIGGER INSERT_E1        -- dipendenza funzionale
NO CASCADE BEFORE INSERT ON E1
REFERENCING NEW AS N
FOR EACH ROW MODE DB2SQL
WHEN (EXISTS (SELECT *
                    FROM E1
                    WHERE E1.A = N.A
                    AND E1.B <> N.B ))
SIGNAL SQLSTATE '70002' ('Violata la dipendenza funzionale A -> B!')@
```