Sistemi Informativi T
7 luglio 2014
Risoluzione

Tempo a disposizione: 2:30 ore

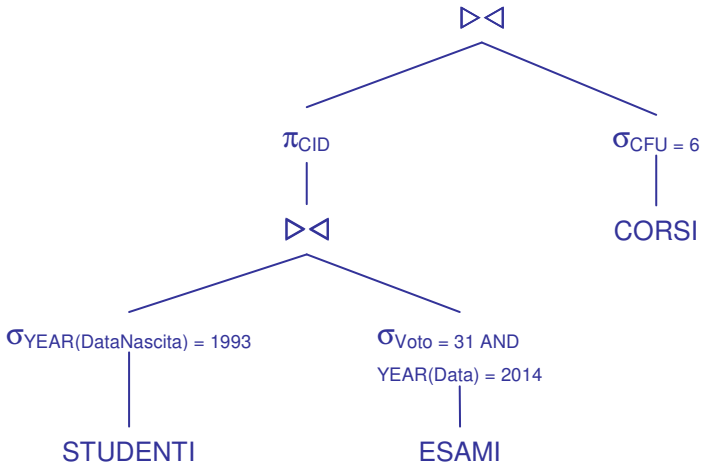
1) Algebra relazionale (3 punti totali):

Date le seguenti relazioni:

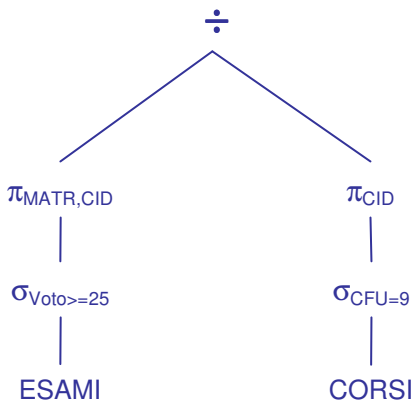
```
STUDENTI(MATR,Nome,Cognome,DataNascita);
CORSI(CID,Nome,Anno,CFU);
ESAMI(MATR,CID,Data,Voto),
      MATR REFERENCES STUDENTI,
      CID REFERENCES CORSI;
--
-- Voto ha valori compresi tra 18 e 31 (31 = 30 e lode)
-- CFU e' un intero positivo
```

si scrivano in algebra relazionale le seguenti interrogazioni:

1.1) **[1 p.]** I dati dei corsi da 6 CFU per cui esiste almeno uno studente nato nel 1993 che ha preso un 30 e lode nel 2014



1.2) **[2 p.]** I numeri di matricola degli studenti che hanno sostenuto tutti gli esami dei corsi da 9 CFU prendendo sempre almeno 25



Sistemi Informativi T
7 luglio 2014
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 numeri di matricola degli studenti che hanno sostenuto tutti gli esami dei corsi da 9 CFU prendendo sempre almeno 25

```
SELECT   E.MATR
FROM     CORSI C, ESAMI E
WHERE    E.CID = C.CID
AND      C.CFU = 9
AND      E.VOTO >= 25
GROUP BY E.MATR
HAVING   COUNT(*)=(  SELECT COUNT(*)
                     FROM CORSI C1
                     WHERE C1.CFU = 9 )
```

2.2) **[3 p.]** Per ogni anno di corso, i numeri di matricola degli studenti che, negli esami di quell'anno, hanno la più alta media dei voti

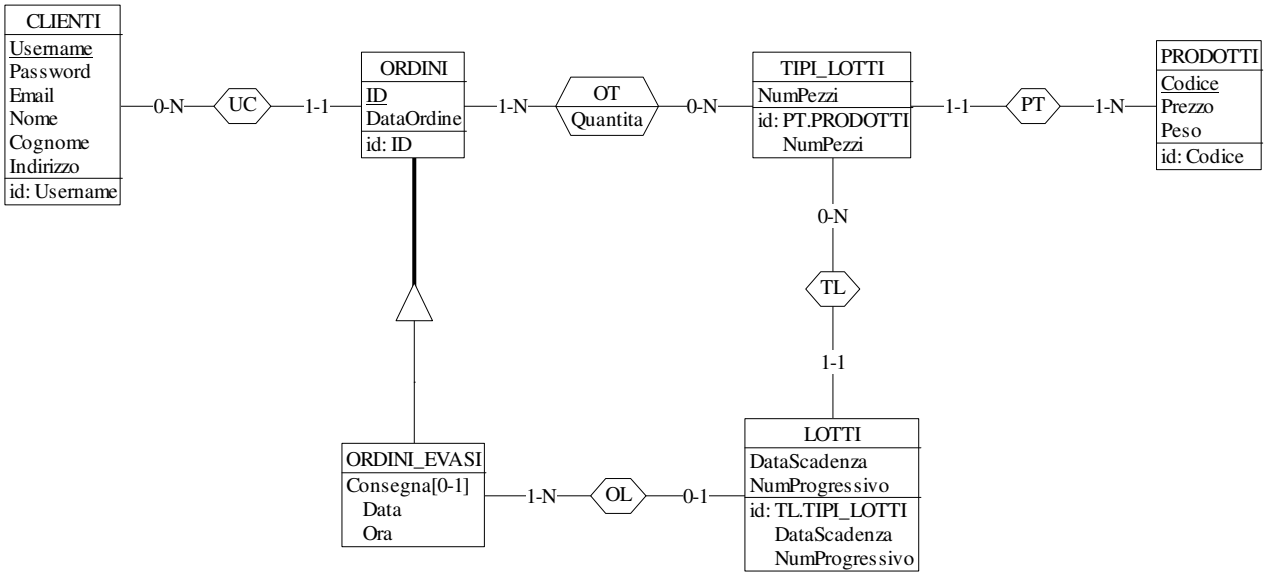
```
WITH MEDIEANNO (MATR,Anno,Media) AS (
   SELECT   E.MATR, C.Anno, AVG(CAST(E.Voto AS DEC(4,2)))
   FROM     CORSI C, ESAMI E
   WHERE    E.CID = C.CID
   GROUP BY E.MATR, C.Anno       )

SELECT   M.Anno,M.MATR
FROM     MEDIEANNO M
WHERE    M.Media = ( SELECT MAX(M1.Media)
                     FROM  MEDIEANNO M1
                     WHERE M1.Anno = M.Anno )
```

Sistemi Informativi T
7 luglio 2014
Risoluzione

3) Progettazione concettuale (6 punti)

La ditta MANGIABEN (MB) distribuisce prodotti alimentari freschi sul territorio nazionale. I prodotti hanno un codice univoco, un prezzo e un peso unitario, e vengono consegnati solo in lotti. Ogni lotto è composto da pezzi di un solo prodotto; per ogni prodotto possono esserci diversi lotti, che si differenziano per il numero di pezzi contenuti (ad es. il latte WING viene venduto in lotti da 12 e 24 pezzi). I pezzi di uno specifico lotto hanno tutti la stessa data di scadenza. Lotti dello stesso tipo e con la stessa data di scadenza si differenziano tra loro per un numero progressivo (ad es. latte WING da 12, scadenza 14/07/2014, numero 25).
I clienti (registrati sul sito della MB con username, password, email e dati anagrafici) possono ordinare ogni volta diversi prodotti, che vengono consegnati sempre assieme. Per ogni ordine si mantiene l'elenco dei prodotti acquistati, il numero e tipo di lotti relativi e la data dell'ordine. Quando l'ordine viene evaso, si registrano i lotti specifici consegnati, e alla consegna la data e l'ora della consegna stessa. La MB garantisce la consegna di tutti i prodotti entro 5 giorni dalla data di scadenza.

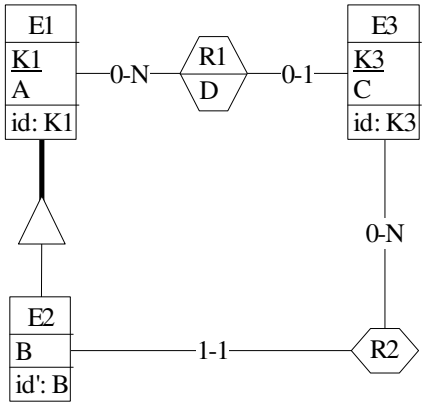


Commenti:

- L'esercizio si risolve agevolmente se si individua il pattern LIBRO-COPIA_LIBRO visto a lezione, qui rappresentato dalle entità TIPI_LOTTI e LOTTI (TIPI_LOTTI è il concetto astratto, che si istanzia nell'entità LOTTI in qualcosa di concreto caratterizzato da DataScadenza e NumProgressivo).
- Non è rappresentabile il vincolo che impone la consistenza dei LOTTI associati a un ordine evaso con i relativi TIPI_LOTTI ordinati, così come non si può ovviamente imporre il vincolo dei 5 giorni citato nel testo.

Sistemi Informativi T
7 luglio 2014
Risoluzione

- **Progettazione logica (6 punti totali)**



Dato lo schema concettuale in figura e considerando che:

a) tutti gli attributi sono di tipo INT;
b) le entità E1 ed E2 vengono tradotte insieme;
c) le associazioni R1 e R2 non vengono tradotte separatamente;
d) un'istanza di E2 può essere associata tramite R2 a un'istanza di E3 solo se i valori di A e C sono diversi;

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,
TIPO SMALLINT NOT NULL CHECK (TIPO IN (1,2)),          -- 2: istanza anche di E2
B INT,
K3R2 INT,
CONSTRAINT E2 CHECK (
     (TIPO = 1 AND B IS NULL AND K3R2 IS NULL) OR
     (TIPO = 2 AND B IS NOT NULL AND K3R2 IS NOT NULL) )                    );

CREATE TABLE E3 (
K3 INT NOT NULL PRIMARY KEY,
C INT NOT NULL,
K1R1 INT REFERENCES E1,
D INT,
CONSTRAINT R1_DEFINED CHECK (
     (K1R1 IS NULL AND D IS NULL) OR
     (K1R1 IS NOT NULL AND D IS NOT NULL) )                                 );

ALTER TABLE E1
ADD CONSTRAINT FKR2 FOREIGN KEY (K3R2) 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 se necessario il simbolo '@' per terminare gli statement SQL (altrimenti ';')

```
-- Trigger che garantisce l'unicità dei valori di B (chiave con valori nulli). Funziona correttamente anche se l'istanza
--  inserita non è di E2, nel qual caso B è nullo
CREATE TRIGGER B_UNIQUE
BEFORE INSERT ON E1
REFERENCING NEW AS N
FOR EACH ROW
WHEN (EXISTS  ( SELECT * FROM E1
                    WHERE N.B = E1.B   ) )
SIGNAL SQLSTATE '70001' ('I valori di B non possono essere duplicati!');


-- Trigger che garantisce il rispetto del vincolo di cui al punto d). Anche in questo caso se l'istanza inserita
-- non appartiene a E2 il trigger funziona correttamente
CREATE TRIGGER PUNTO_D
BEFORE INSERT ON E1
REFERENCING NEW AS N
FOR EACH ROW
WHEN (EXISTS ( SELECT *  FROM E3
                    WHERE    N.K3R2 = E3.K3
                    AND      N.A = E3.C        ) )
SIGNAL SQLSTATE '70002' ('La tupla referenziata non può avere C uguale ad A!');
```