Sistemi Informativi T
13 febbraio 2023
Risoluzione

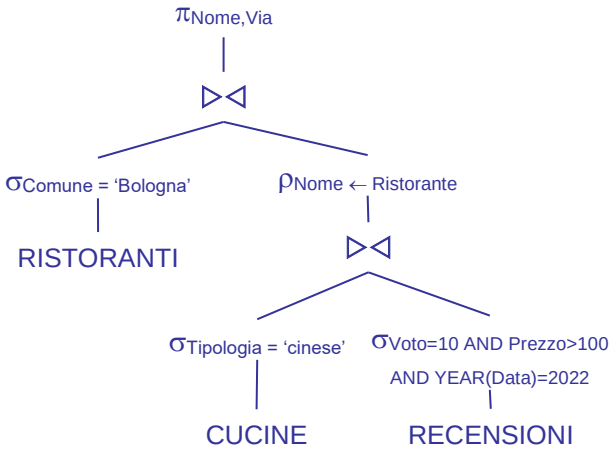
1) Algebra relazionale (3 punti totali):

Date le seguenti relazioni:

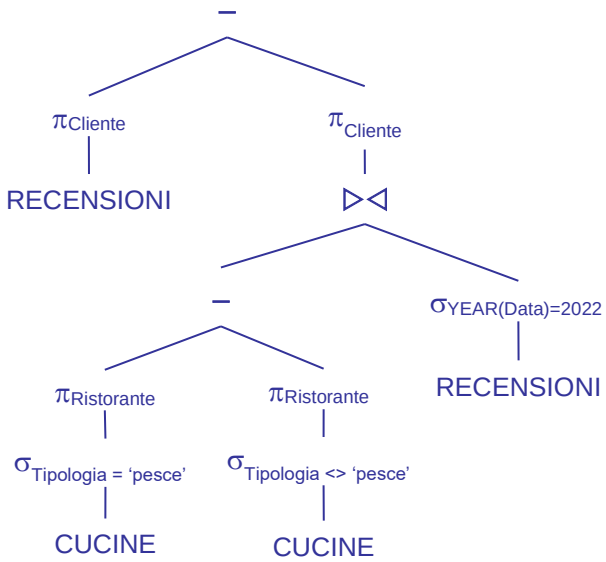
```
RISTORANTI(Nome,Via,Comune);
CUCINE(Ristorante,Tipologia),
      Ristorante REFERENCES RISTORANTI;
RECENSIONI(Ristorante,Cliente,Data,NumPersone,Prezzo,Voto),
      Ristorante REFERENCES RISTORANTI;
-- NumPersone è di tipo INT > 0.
-- Prezzo è di tipo DEC(6,2): totale pagato per NumPersone
-- Voto è di tipo INT, valori da 1 a 10.
-- Tipologia: pizza, pesce, carne, cinese, ecc.
```

si esprimano in algebra relazionale le seguenti interrogazioni:

1.1) [1 p.] Nome e via dei ristoranti di cucina cinese a Bologna che nel 2022 hanno ricevuto almeno un 10 da un cliente che ha pagato più di 100€



1.2) [2 p.] I clienti che nel 2022 non hanno mai recensito ristoranti che fanno solo pesce



La prima differenza trova i ristoranti che fanno solo pesce, e quindi l'operando destro della seconda differenza sono i clienti che nel 2022 hanno recensito almeno un ristorante di tale tipo.
La condizione Tipologia = 'pesce' si può omettere, qui è lasciata solo per maggior chiarezza.

Sistemi Informativi T
13 febbraio 2023
Risoluzione

2) SQL (5 punti totali)

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

2.1) [2 p.] Per ogni ristorante che propone sia carne che pesce e con almeno 2 recensioni, la media dei voti, ordinando per comune e quindi per media decrescente

```
SELECT   RC.RISTORANTE, R.COMUNE, DEC(AVG(RC.VOTO*1.0),4,2) AS MEDIA_VOTI
FROM     RISTORANTI R , RECENSIONI RC
WHERE    R.NOME = RC.RISTORANTE
AND      R.NOME IN ( SELECT C1.RISTORANTE
                     FROM   CUCINE C1, CUCINE C2
                     WHERE  C1.RISTORANTE = C2.RISTORANTE
                     AND    C1.TIPOLOGIA = 'pesce'
                     AND    C2.TIPOLOGIA = 'carne')
GROUP BY RC.RISTORANTE, R.COMUNE
HAVING COUNT(*) >= 2             -- almeno 2 recensioni
ORDER BY R.COMUNE, MEDIA_VOTI DESC;
```

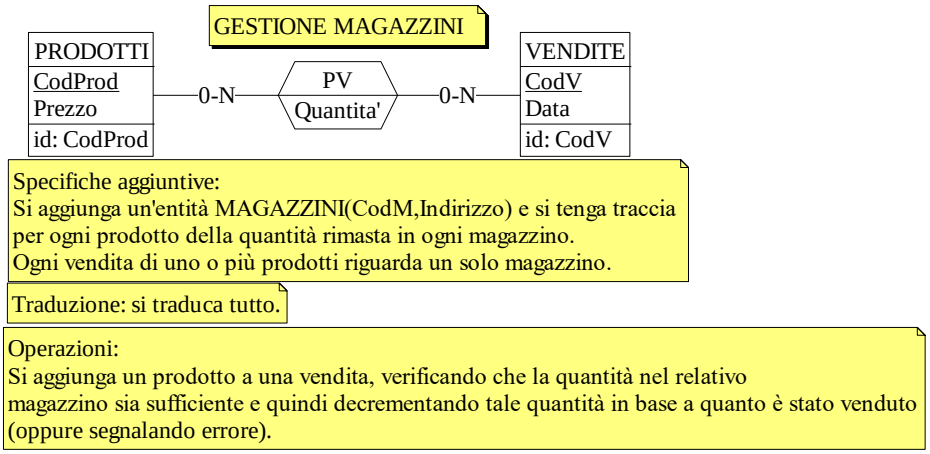
2.2) [3 p.] Considerando solo le recensioni di clienti che hanno recensito almeno 2 ristoranti diversi, per ogni comune il ristorante che ha il miglior rapporto qualità/prezzo, calcolato come la media di voto/(prezzo per persona)

```
WITH
RECE_VALIDE(RISTORANTE,COMUNE,VOTO,PREZZO_PERSONA) AS
(SELECT RC.RISTORANTE, R.COMUNE, RC.VOTO, RC.PREZZO/RC.NUMPERSONE
 FROM   RISTORANTI R , RECENSIONI RC
 WHERE  R.NOME = RC.RISTORANTE
 AND    RC.CLIENTE IN ( SELECT  RC1.CLIENTE
                        FROM    RECENSIONI RC1
                        GROUP BY RC1.CLIENTE
                        HAVING  COUNT(DISTINCT RC1.RISTORANTE) >= 2)
),
QUALITA_PREZZO(RISTORANTE,COMUNE,QP) AS
(SELECT  RV.RISTORANTE,RV.COMUNE,
         AVG(DEC(RV.VOTO,4,2)/RV.PREZZO_PERSONA)
 FROM    RECE_VALIDE RV
 GROUP BY RV.RISTORANTE,RV.COMUNE
)
SELECT  Q.*
FROM    QUALITA_PREZZO Q
WHERE   Q.QP = ( SELECT MAX(Q1.QP)
                 FROM   QUALITA_PREZZO Q1
                 WHERE  Q1.COMUNE = Q.COMUNE );

-- La prima c.t.e. calcola il prezzo medio per ogni recensione valida
-- (notare la forma COUNT(DISTINCT …) per contare il numero distinto
-- di ristoranti recensiti da un cliente), e la seconda il rapporto
-- qualità/prezzo.
.
```

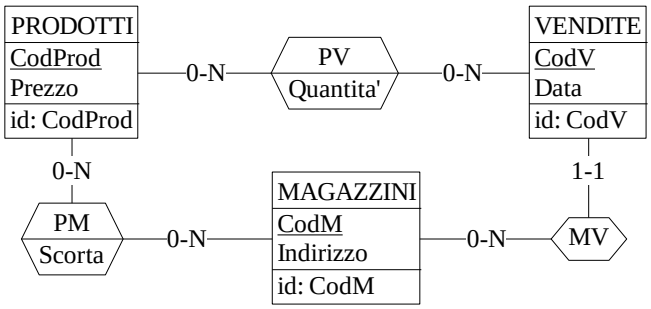
Sistemi Informativi T
13 febbraio 2023
Risoluzione

3) Modifica di schema E/R e del DB (6 punti totali)

Dato il file ESE3.lun fornito, in cui è presente lo schema ESE3-input in figura:



3.1) [2 p.] Si modifichi ESE3-input secondo le Specifiche aggiuntive;



3.2) [1 p.] Si veda il relativo file .sql

3.3) [3 p.] Si scriva l'istruzione SQL che modifica il DB come da specifiche (usare valori a scelta) e si definiscano i trigger necessari.

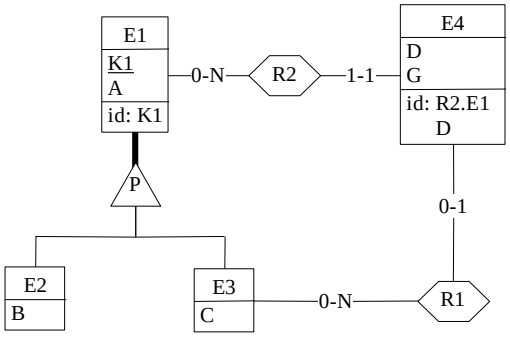
```
INSERT INTO PV VALUES (:CodProdotto,:CodVendita,:Quantità);

CREATE OR REPLACE TRIGGER CHECK_SCORTA
BEFORE INSERT ON PV
REFERENCING NEW AS N
FOR EACH ROW
WHEN (N.Quantita > (        SELECT  PM.Scorta
                            FROM    PM, VENDITE V
                            WHERE   PM.CodProd = N.CodProd
                            AND     PM.CodM = V.CodM
                            AND     V.CodV = N.CodV         ))
SIGNAL SQLSTATE '70001' ('Scorta insufficiente!');

CREATE OR REPLACE TRIGGER AGGIORNA_SCORTA
AFTER INSERT ON PV
REFERENCING NEW AS N
FOR EACH ROW
UPDATE      PM
SET         PM.Scorta = PM.Scorta - N.Quantita
WHERE               PM.CodProd = N.CodProd
AND                 PM.CodM = ( SELECT V.CodM
                                FROM    VENDITE V
                                WHERE V.CodV = N.CodV  ) ;
```

Sistemi Informativi T
13 febbraio 2023
Risoluzione

Progettazione logica (6 punti totali)

Dato lo schema concettuale in figura e considerando che:

a) le entità E1, E2 ed E3 vengono tradotte assieme;
b) nessuna associazione viene tradotta separatamente;
c) le istanze di E4 identificate esternamente dalla stessa istanza di E1 hanno valori di G la cui somma non supera il valore del corrispondente A;



4.1) [3 p.] Si progettino gli opportuni schemi relazionali e si definiscano tali schemi mediante uno script SQL compatibile con DB2

```
-- il tipo degli attributi non è necessariamente INT

CREATE TABLE E1 (
K1          INT NOT NULL PRIMARY KEY,
A           INT NOT NULL,
TIPO23      SMALLINT NOT NULL CHECK (TIPO23 IN (2,3)), -- 2:istanza di E2, 3: istanza di E3
B           INT,
C           INT,
CONSTRAINT E2E3 CHECK ((TIPO23 = 2 AND B IS NOT NULL AND C IS NULL) OR
                       (TIPO23 = 3 AND B IS NULL AND C IS NOT NULL))    );

CREATE TABLE E4 (
K1          INT NOT NULL REFERENCES E1,
D           INT NOT NULL,
G           INT NOT NULL,
K1R1        INT REFERENCES E1,
PRIMARY KEY (K1,D)                  );
```

4.2) [3 p.] Per i vincoli non esprimibili a livello di schema si predispongano opportuni trigger che evitino **inserimenti di singole tuple non corrette**

```
-- Trigger che garantisce che R1 referenzi un'istanza di E3
CREATE OR REPLACE TRIGGER R1_E3
BEFORE INSERT ON E4
REFERENCING NEW AS N
FOR EACH ROW
WHEN ( EXISTS ( SELECT *
                FROM E1
                WHERE N.K1R1 = E1.K1
                AND E1.TIPO23 = 2 ) )
SIGNAL SQLSTATE '70001' ('La tupla referenzia una tupla che non appartiene a E3!');

CREATE OR REPLACE TRIGGER PUNTO_C
BEFORE INSERT ON E4
REFERENCING NEW AS N
FOR EACH ROW
WHEN (    (       SELECT E1.A FROM E1 WHERE E1.K1 = N.K1  ) <
          (       SELECT N.G + COALESCE(SUM(E4.G),0)
                  FROM   E4
                  WHERE E4.K1 = N.K1  ) )
SIGNAL SQLSTATE '70002' ('La somma dei valori di G supera il valore di A! ');

-- COALESCE(SUM(E4.G),0) serve per gestire il caso in cui la prima tupla con un certo valore di K1
-- abbia G > A
```