Sistemi Informativi T
11 settembre 2025
Risoluzione

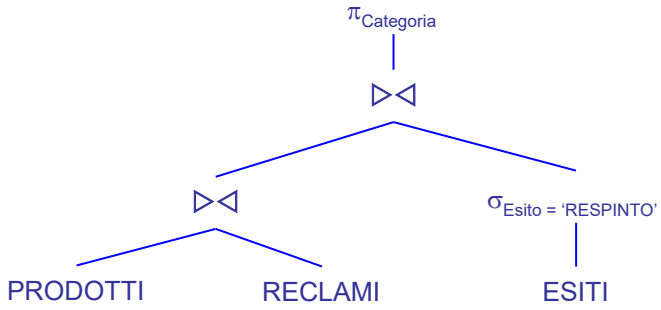
1) Algebra relazionale (3 punti totali):

Date le seguenti relazioni:

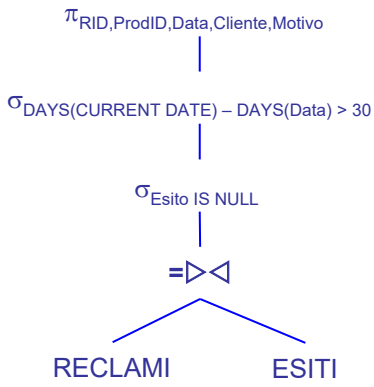
```
PRODOTTI(ProdID,Categoria,Prezzo);
RECLAMI(RID,ProdID,Data,Cliente,Motivo),
      ProdID REFERENCES PRODOTTI;
ESITI(RID,DataEsito,Esito,Rimborso*),
      RID REFERENCES RECLAMI;
-- DataEsito = la data in cui si è deciso come trattare il reclamo
-- Esito = descrive se e come il reclamo è stato accolto
-- Se Esito = 'RIMBORSO', allora l'attributo Rimborso riporta l'importo
--   rimborsato al cliente (minore o uguale del prezzo del prodotto),
--   altrimenti Rimborso è NULL
-- Prezzo e Rimborso sono di tipo DEC(6,2)
```

si esprimano in algebra relazionale le seguenti interrogazioni:

1.1) **[1 p.]** Le categorie per le quali esiste almeno un prodotto con un reclamo con esito ‘RESPINTO’



1.2) **[2 p.]** I dati dei reclami per i quali, considerando la data odierna, sono passati più di 30 giorni senza che ci sia ancora un esito



Sistemi Informativi T
11 settembre 2025
Risoluzione

2) SQL (5 punti totali)

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

2.1) [2 p.] I dati dei prodotti con prezzo maggiore di 50€ per i quali il numero di reclami rimborsati è maggiore di quelli non rimborsati (con esito definito o meno)

```
SELECT   P.ProdID,P.Categoria,P.Prezzo
FROM     PRODOTTI P, RECLAMI R LEFT JOIN ESITI E ON (R.RID = E.RID)
WHERE    P.ProdID = R.ProdID
AND      P.Prezzo > 50
GROUP BY P.ProdID,P.Categoria,P.Prezzo
HAVING 2*COUNT(E.Rimborso) > COUNT(*);

-- La condizione nella clausola HAVING equivale a
-- COUNT(E.Rimborso) > COUNT(*)-COUNT(E.Rimborso), in cui il lato sinistro
-- è pari al numero di reclami rimborsati e il lato destro a quelli
-- non rimborsati. Questi ultimi, grazie al left join, includono anche
-- quelli ancora senza esito
```

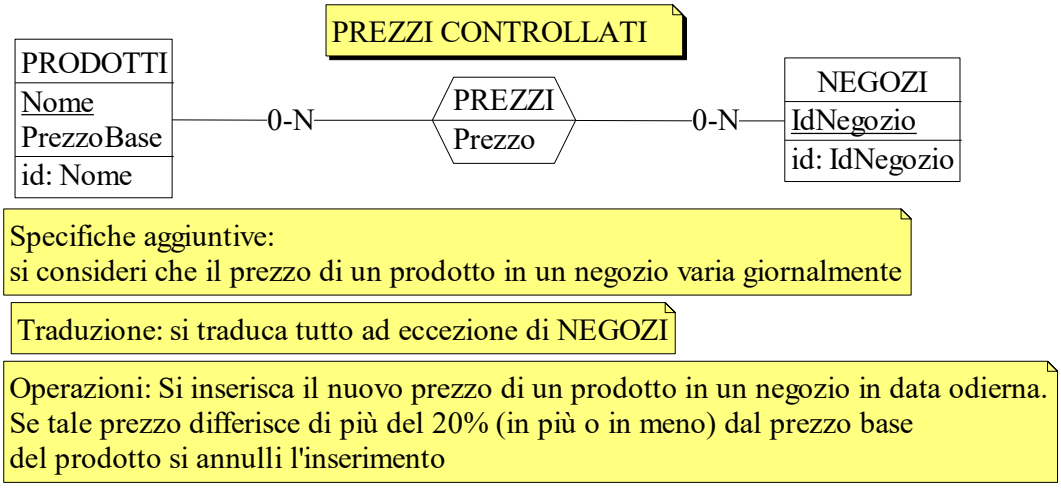
2.2) [3 p.] Per ogni categoria il prodotto per il quale è passato il numero minimo di giorni (≥ 0) tra un reclamo e l'altro

```
WITH DIFF_RECLAMI (Categoria, ProdID,NumGiorni) AS (
   SELECT P.Categoria, R1.ProdID, DAYS(R2.Data)-DAYS(R1.Data)
   FROM   PRODOTTI P, RECLAMI R1, RECLAMI R2
   WHERE  P.ProdID = R1.ProdID
   AND    R1.ProdID = R2.ProdID
   AND    R1.RID != R2.RID
   AND    R1.Data <= R2.Data
)
SELECT   D.*
FROM     DIFF_RECLAMI D
WHERE    D.NumGiorni = ( SELECT MIN(D1.NumGiorni)
                         FROM   DIFF_RECLAMI D1
                         WHERE  D.Categoria = D1.Categoria ) ;
```

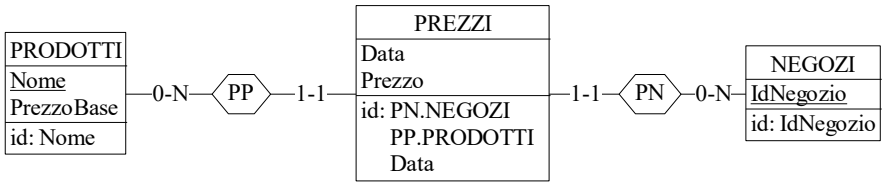
Sistemi Informativi T
11 settembre 2025
Risoluzione

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 copi lo schema ESE3-input in uno schema ESE3-modificato e si modifichi quest'ultimo secondo le Specifiche aggiuntive;



3.2) [1 p.] Si copi lo schema modificato in uno schema ESE3-tradotto. Mediante il comando Transform/Quick SQL, si traduca la parte di schema specificata, modificando lo script SQL in modo da essere compatibile con DB2 e permettere l'esecuzione del punto successivo, ed eventualmente aggiungendo quanto richiesto dalle Specifiche aggiuntive;
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.

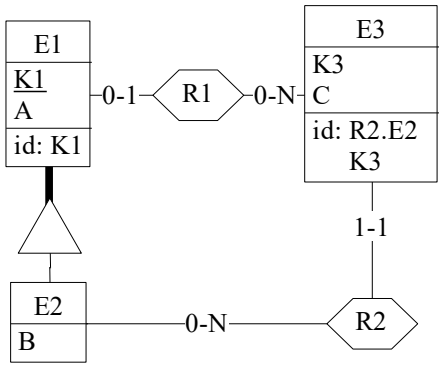
```
CREATE OR REPLACE TRIGGER CONTROLLO_PREZZO
BEFORE INSERT ON PREZZI
REFERENCING NEW AS N
FOR EACH ROW
WHEN ( (ABS(N.Prezzo - (SELECT PrezzoBase FROM PRODOTTI
                                WHERE  Nome = N.Nome)) /
          (SELECT PrezzoBase FROM PRODOTTI WHERE  Nome = N.Nome))
             > 0.2 )
SIGNAL SQLSTATE '70001' ('Prezzo non valido!')          ;

INSERT INTO PREZZI(Nome,IdNegozio,Data,Prezzo)
VALUES(;unProdotto,:negozio,CURRENT DATE,:prezzo) ;
```

Sistemi Informativi T
11 settembre 2025
Risoluzione

4) Progettazione logica (6 punti totali)

Dato lo schema concettuale in figura e considerando che:



a) le entità E1 ed E2 vengono tradotte assieme;
b) nessuna associazione viene tradotta separatamente;
c) un'istanza di E1 non è mai associata tramite R1 a un'istanza di E3 identificata esternamente da un'istanza di E2 con B > 10;

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,
K3          INT,
K1E2        INT,
TIPO12      SMALLINT NOT NULL CHECK (TIPO12 IN (1,2)),   -- se 2 è istanza anche di E2
B           INT
CONSTRAINT E2 CHECK ((TIPO12 = 1 AND B IS NULL) OR (TIPO12 = 2 AND B IS NOT NULL),
CONSTRAINT FK_DEFINED CHECK ((K3 IS NOT NULL AND K1E2 IS NOT NULL) OR
                             (K3 IS NULL AND K1E2 IS NULL)                    );

CREATE TABLE E3 (
K3          INT NOT NULL,
K1E2        INT NOT NULL REFERENCES E1,
C           INT NOT NULL,
PRIMARY KEY (K3,K1E2)            );

ALTER TABLE E1
ADD CONSTRAINT FK_R1 FOREIGN KEY (K3,K1E2) REFERENCES E3        ;
```

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 R2 referenzi un'istanza di E2
CREATE OR REPLACE TRIGGER R2_E2
BEFORE INSERT ON E3
REFERENCING NEW AS N
FOR EACH ROW
WHEN ( EXISTS (    SELECT *
                   FROM E1
                   WHERE N.K1E2 = E1.K1
                   AND E1.TIPO12 = 1 ) )
SIGNAL SQLSTATE '70001' ('La tupla referenzia una tupla che non appartiene a E2!');

-- Si noti che non è necessario accedere a E3, in quanto il valore di N.K1E2 corrisponde senz'altro
-- a un valore di E3.K1E2 (vincolo di foreign key), e quindi testare (N.K3,N.K1E2) = (E3.K3,E3.K1E2)
-- AND E3.K1E2=E2.K1 equivale al più semplice N.K1E2 = E2.K1
CREATE TRIGGER PUNTO_C
BEFORE INSERT ON E1
REFERENCING NEW AS N
FOR EACH ROW
WHEN ( EXISTS (   SELECT *
                  FROM    E1 E2
                  WHERE  N.K1E2 = E2.K1
                  AND     E2.B > 10 ))
SIGNAL SQLSTATE '70002' ('La tupla inserita in E1 non rispetta il vincolo del punto c)! ');
```