Sistemi Informativi T
7 luglio 2025
Risoluzione

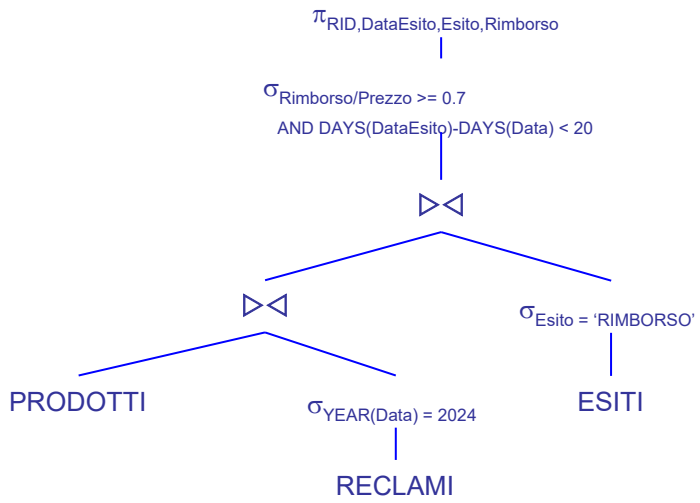
1) Algebra relazionale (3 punti totali):

Date le seguenti relazioni:

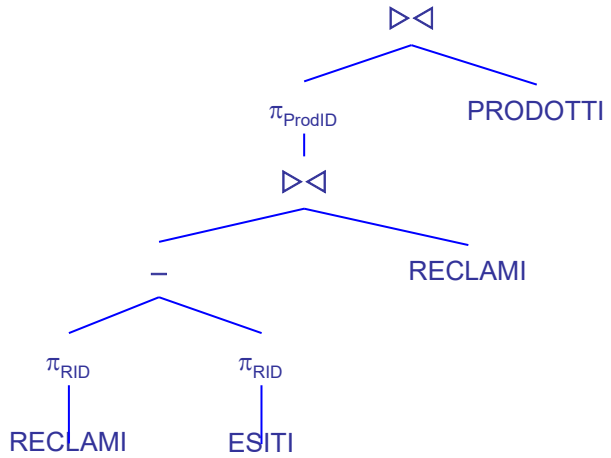
```
PRODOTTI(ProdID,Categoria,Prezzo);
RECLAMI(RID,ProdID,Data,Cliente,Motivo),
      ProdID REFERENCES PRODOTTI;
ESITI(RID,DataEsito,Esito,Rimborso*),
      RID REFERENCES RECLAMI;
-- DataEsito = la data in cui si è deciso come trattare il reclamo
-- Esito = descrive se e come il reclamo è stato accolto
-- 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.]** I dati degli esiti di reclami del 2024 che sono stati rimborsati per almeno il 70% del prezzo del prodotto in meno di 20 giorni



1.2) **[2 p.]** I dati dei prodotti con almeno un reclamo senza esito definito



La differenza trova i reclami senza esito, il successivo join serve a recuperare i relativi codici dei prodotti.

Sistemi Informativi T
7 luglio 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.] Per ogni categoria, la media del rapporto Rimborso/Prezzo per i soli prodotti con più di un reclamo

```
SELECT   P.Categoria, DEC(AVG(E.Rimborso/P.Prezzo),4,3)
FROM     PRODOTTI P, RECLAMI R, ESITI E
WHERE    P.ProdID = R.ProdID
AND      R.RID = E.RID
AND      P.ProdID IN ( SELECT R1.ProdID
                       FROM   RECLAMI R1
                       GROUP BY R1.ProdID
                       HAVING COUNT(*) > 1 )
GROUP BY P.Categoria;
```

2.2) [3 p.] La categoria per cui i reclami con esito definito sono stati in media più veloci

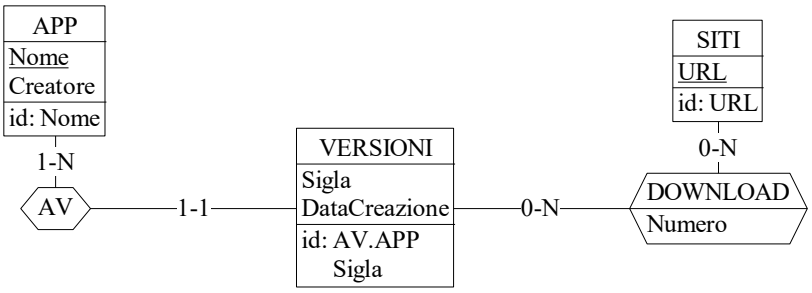
```
WITH TEMPI_ESITI (Categoria,MediaGiorni) AS (
   SELECT P.Categoria, AVG(DEC(DAYS(E.DataEsito)-DAYS(R.Data),6,3))
   FROM   PRODOTTI P, RECLAMI R, ESITI E
   WHERE  P.ProdID = R.ProdID
   AND    R.RID = E.RID
   GROUP BY P.Categoria
)
SELECT   T.*
FROM     TEMPI_ESITI T
WHERE    T.MediaGiorni = ( SELECT MIN(T1.MediaGiorni)
                           FROM   TEMPI_ESITI T1 ) ;

-- E' necessario eseguire il CAST della differenza di giorni prima di
-- calcolare la media
```

Sistemi Informativi T
7 luglio 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

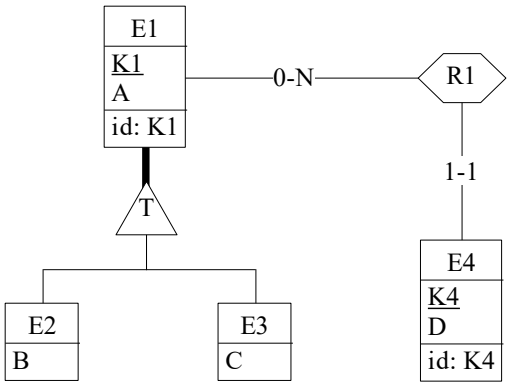
3.3) [3 p.] Si scriva l'istruzione SQL che modifica il DB come da specifiche (usare valori a scelta) e si definiscano i trigger necessari.

```
CREATE OR REPLACE TRIGGER INIT_NUM_DOWNLOAD
AFTER INSERT ON VERSIONI
REFERENCING NEW AS N
FOR EACH ROW
INSERT INTO DOWNLOAD(Nome,Sigla,URL)  -- Numero ha valore di default 0
SELECT DISTINCT N.Nome,N.Sigla,URL
FROM    DOWNLOAD D
WHERE D.Nome = N.Nome    ;

INSERT INTO VERSIONI(Nome,Sigla,DataCreazione)
VALUES(:nomeApp,:siglaVersione,CURRENT DATE) ;

-- Il DISTINCT serve a evitare di inserire lo stesso URL più volte, che violerebbe il vincolo di primary key
```

Sistemi Informativi T
7 luglio 2025
Risoluzione

4) Progettazione logica (6 punti totali)

Dato lo schema concettuale in figura e considerando che:



a) la gerarchia viene tradotta operando un collasso verso il basso;
b) l'associazione R1 non viene tradotta separatamente;
c) un'istanza di E4 non è mai associata a un'istanza di E1 che è istanza sia di E2 che di E3;

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 E2 (
K1          INT NOT NULL PRIMARY KEY,
A           INT NOT NULL,
B           INT NOT NULL                    );

CREATE TABLE E3 (
K1          INT NOT NULL PRIMARY KEY,
A           INT NOT NULL,
C           INT NOT NULL                    );

CREATE TABLE E4 (
K4          INT NOT NULL PRIMARY KEY,
D           INT NOT NULL,
K1E2        INT REFERENCES E2,
K1E3        INT REFERENCES E3,
CONSTRAINT R1_DEFINED CHECK ((K1E2 IS NULL AND K1E3 IS NOT NULL) OR
                             (K1E2 IS NOT NULL AND K1E3 IS NULL))    );
```

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**

```
-- Il vincolo al punto c) può essere violato solo inserendo in E4; la violazione può avvenire in due casi,
-- dipendentemente da quale delle due FK è non nulla

CREATE TRIGGER PUNTO_C
BEFORE INSERT ON E4
REFERENCING NEW AS N
FOR EACH ROW
WHEN ( EXISTS (   SELECT *     -- per il caso in cui N.K1E2 non è nulla
                  FROM    E3
                  WHERE  N.K1E2 = E3.K1  )
          OR
       EXISTS (   SELECT *     -- per il caso in cui N.K1E3 non è nulla
                  FROM    E2
                  WHERE  N.K1E3 = E2.K1  ) )
SIGNAL SQLSTATE '70001' ('La tupla inserita in E4 non rispetta il vincolo del punto c)! ');
```