

Sistemi Informativi T
18 febbraio 2022
Risoluzione

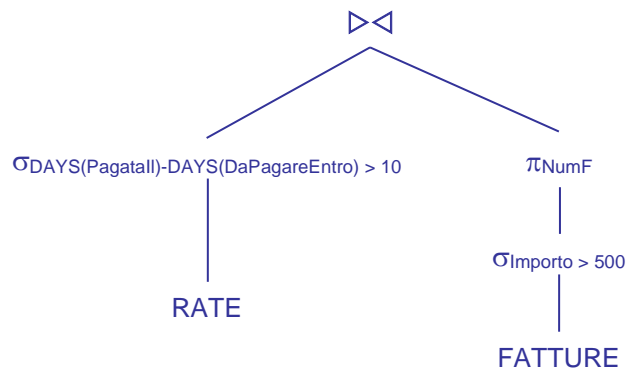
1) Algebra relazionale (3 punti totali):

Date le seguenti relazioni:

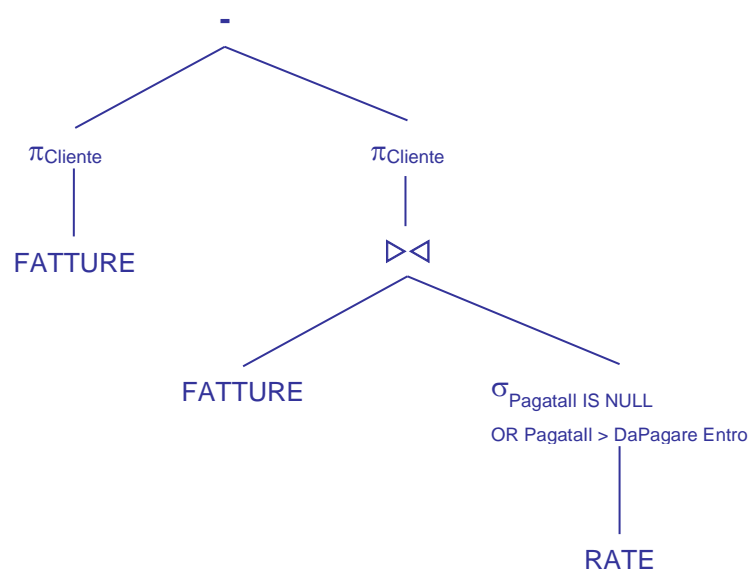
```
FATTURE (NumF, Data, Importo, Cliente, NumeroRate);  
RATE (NumF, NumRate, ImportoRate, DaPagareEntro, PagataIl*),  
      NumF REFERENCES FATTURE;  
-- Importo e ImportoRate sono di tipo DEC(8,2).  
-- NumeroRate (>1) è il numero di rate da pagare per la fattura.  
-- NumRate è un intero che va da 1 a Numero Rate e distingue le  
-- diverse rate di una stessa fattura.  
-- La somma di ImportoRate corrisponde all'Importo della fattura.  
-- PagataIl è NULL se la rata non è stata ancora pagata.
```

si esprimano in algebra relazionale le seguenti interrogazioni:

1.1) [1 p.] I dati delle rate pagate con più di 10 giorni di ritardo relative a fatture di importo superiore a 500€



1.2) [2 p.] I clienti che hanno pagato tutte le rate delle loro fatture, e sempre senza ritardi



L'operando destro della differenza trova i clienti con almeno una rata non ancora pagata o con almeno un pagamento in ritardo

Sistemi Informativi T
18 febbraio 2022
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 cliente la percentuale di rate pagate, considerando solo le fatture non completamente pagate

```
SELECT    F.Cliente,
          DEC(COUNT(R.PagataIl)*100.0/COUNT(*),4,2) || '%' AS PERC_PAGATE
FROM      FATTURE F, RATE R
WHERE     F.NumF = R.NumF
AND       F.NumF IN ( SELECT R1.NumF
                     FROM   RATE R1
                     WHERE  R1.PagataIl IS NULL )

GROUP BY F.Cliente ;
```

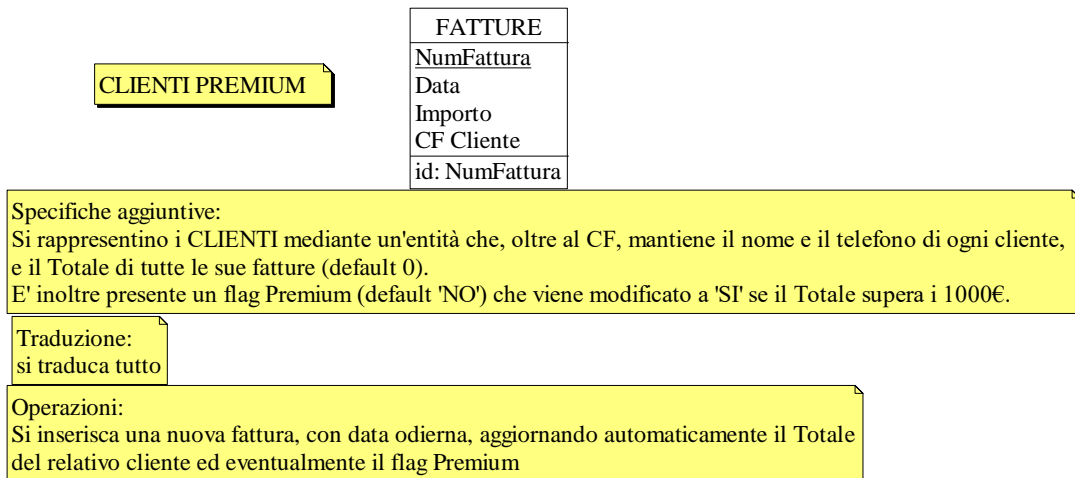
- 2.2) [3 p.]** I dati della fattura in cui, considerando solo le rate pagate in ritardo, la media dei giorni di ritardato pagamento è stata massima

```
WITH RITARDI (NumF,RitardoMedio) AS (
    SELECT R.NumF,
           DEC(AVG((DAYS(R.PagataIl)-DAYS(R.DaPagareEntro))*1.0),6,2)
    FROM   RATE R
    WHERE  R.PagataIl > R.DaPagareEntro
    GROUP BY R.NumF
)
SELECT  F.*,R.RitardoMedio
FROM    FATTURE F, RITARDI R
WHERE   F.NumF = R.NumF
AND     R.RitardoMedio = ( SELECT MAX(R1.RitardoMedio)
                        FROM   RITARDI R1 )
;

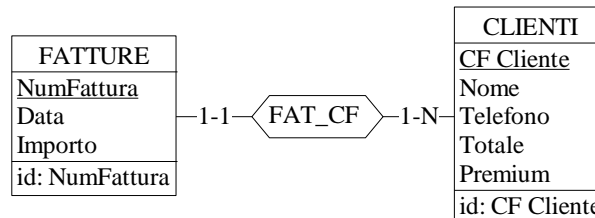
-- La c.t.e. calcola il ritardo medio per ogni fattura.
-- Si noti che se una rata non è stata pagata viene automaticamente
-- esclusa dal calcolo della media perché il predicato
-- R.PagataIl > R.DaPagareEntro non è soddisfatto.
-- E' fondamentale eseguire il casting prima di calcolare la media
```

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 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 AGGIORNA_TOTALE
AFTER INSERT ON FATTURE
REFERENCING NEW AS N
FOR EACH ROW
UPDATE CLIENTI
SET      Totale = Totale + N.Importo
WHERE    CF_Cliente = N.CF_Cliente;
```

```
CREATE OR REPLACE TRIGGER AGGIORNA_FLAG
AFTER UPDATE OF Totale ON CLIENTI
REFERENCING NEW AS N
FOR EACH ROW
WHEN (N.Totale > 1000 AND N.Premium = 'NO')
UPDATE CLIENTI
SET      Premium = 'SI'
WHERE    CF_Cliente = N.CF_Cliente;
```

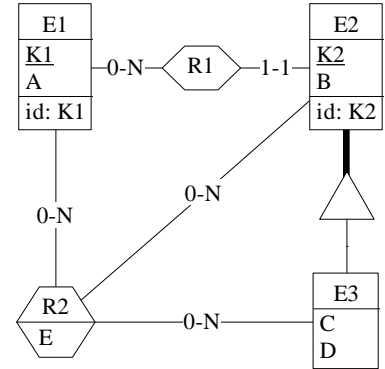
```
INSERT INTO FATTURE VALUES
(:numFattura, CURRENT DATE, :importo, :CFcliente);
```

Sistemi Informativi T
18 febbraio 2022
Risoluzione

4) Progettazione logica (6 punti totali)

Dato lo schema concettuale in figura e considerando che:

- le entità E2 ed E3 vengono tradotte insieme;
- l'associazione R1 non viene tradotta separatamente;
- le istanze di E1 che partecipano a R2 hanno valori di A tali che $A > E$;



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
);
```

```
CREATE TABLE E2 (
K2          INT NOT NULL PRIMARY KEY,
B          INT NOT NULL,
K1R1       INT NOT NULL REFERENCES E1,
TIPO3      SMALLINT NOT NULL CHECK (TIPO3 IN (2,3)), -- TIPO3 = 3 se appartiene anche a E3
C          INT,
D          INT,
CONSTRAINT E3 CHECK ((TIPO3 = 2 AND C IS NULL AND D IS NULL)
OR (TIPO3 = 3 AND C IS NOT NULL AND D IS NOT NULL))
);
```

```
CREATE TABLE R2 (
K1          INT NOT NULL REFERENCES E1,
K2E2       INT NOT NULL REFERENCES E2,
K2E3       INT NOT NULL REFERENCES E2,
E          INT NOT NULL,
PRIMARY KEY (K1,K2E2,K2E3)
);
```

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**

```
CREATE TRIGGER K2E3_REFERENCES_E3
BEFORE INSERT ON R2
REFERENCING NEW AS N
FOR EACH ROW
WHEN ( NOT EXISTS ( SELECT *
                    FROM   E2
                    WHERE  N.K2E3 = E2.K2
                    AND    E2.TIPO3 = 3 ) )
SIGNAL SQLSTATE '70001' ('La foreign key K2E3 deve referenziare un'istanza di E3!');
```

```
-- Il vincolo al punto c) può essere violato solo inserendo in R2
CREATE TRIGGER PUNTO_C
BEFORE INSERT ON R2
REFERENCING NEW AS N
FOR EACH ROW
WHEN (EXISTS ( SELECT *
               FROM   E1
               WHERE  N.K1 = E1.K1
               AND    N.E >= E1.A ) )
SIGNAL SQLSTATE '70002' ('La tupla di E1 deve avere A > E!');
```