

Sistemi Informativi T
9 gennaio 2025
Risoluzione

1) Algebra relazionale (3 punti totali):

Date le seguenti relazioni:

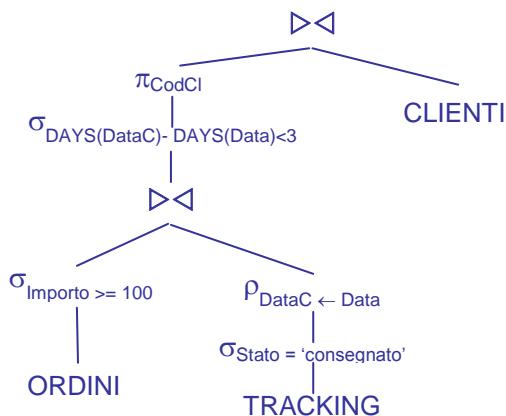
```

CLIENTI (CodCl, Nome, Via, Comune);
ORDINI (IDOrd, CodCl, Importo, Data),
        CodCl REFERENCES CLIENTI;
TRACKING (IDOrd, Stato, Data),
        IDOrd REFERENCES ORDINI;
-- ORDINI.Data è la data in cui è stato fatto l'ordine.
-- Stato indica lo stato dell'ordine, e i possibili valori sono,
-- in sequenza temporale: 'spedito', 'arrivato' e 'consegnato'.
-- Importo è di tipo DEC(6,2).

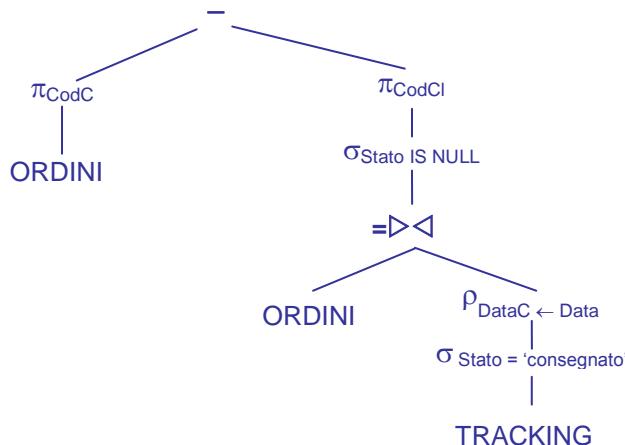
```

si esprimano in algebra relazionale le seguenti interrogazioni:

- 1.1) [1 p.]** I dati dei clienti che hanno fatto almeno un ordine di 100€ o più e che è stato consegnato in meno di 3 giorni



- 1.2) [2 p.]** I codici dei clienti che hanno avuto tutti i loro ordini consegnati



L'operando destro della differenza trova i codici dei clienti con almeno un ordine non consegnato.
L'operando sinistro della differenza usa ORDINI anziché CLIENTI per non restituire anche i clienti che non hanno fatto nessun ordine.

Sistemi Informativi T

9 gennaio 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 comune, considerando solo gli ordini spediti lo stesso giorno in cui è stato fatto l'ordine e che sono stati consegnati, l'importo medio di un ordine

```
SELECT      C.COMUNE, AVG(O.IMPORTO) AS MEDIA_IMPORTO
FROM        CLIENTI C, ORDINI O, TRACKING T
WHERE       C.CODCL = O.CODCL
AND        O.IDORD = T.IDORD
AND        O.DATA = T.DATA          -- spedito lo stesso giorno
AND        T.STATO = 'spedito'
AND        EXISTS ( SELECT *      -- ordine consegnato
                  FROM   TRACKING T2
                  WHERE  T2.IDORD = O.IDORD
                  AND    T2.STATO = 'consegnato')
GROUP BY C.COMUNE;
```

- 2.2) [3 p.] L'identificativo degli ordini per cui la durata di un passaggio di stato (da 'spedito' ad 'arrivato' o da 'arrivato' a 'consegnato') è stata massima (se l'ordine non è stato consegnato il passaggio da 'arrivato' a 'consegnato' non va considerato, idem se un ordine non è 'arrivato')

```
WITH
DURATE_STATI (IDORD,DURATA) AS (
  SELECT T1.IDORD, DAYS(T2.DATA)-DAYS(T1.DATA)
  FROM   TRACKING T1, TRACKING T2
  WHERE  T1.IDORD = T2.IDORD
  AND   ((T2.STATO = 'arrivato' AND T1.STATO = 'spedito')
  OR   (T2.STATO = 'consegnato' AND T1.STATO = 'arrivato')) )
SELECT D1.*
FROM   DURATE_STATI D1
WHERE  D1.DURATA = ( SELECT MAX(D2.DURATA)
                      FROM   DURATE_STATI D2 );
-- La c.t.e. calcola le durate per tutti i passaggi di stato
```

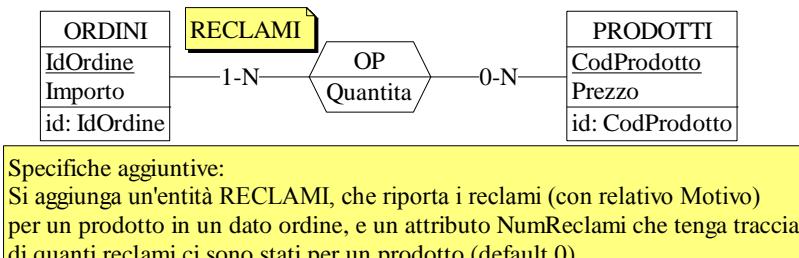
Sistemi Informativi T

9 gennaio 2025

Risoluzione

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

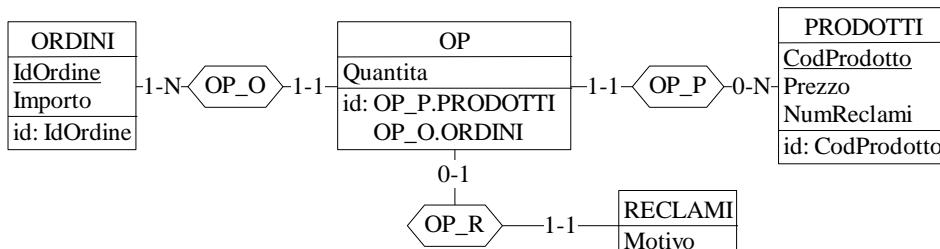
Dato il file ESE3.lun fornito, in cui è presente lo schema ESE3-input in figura:



Traduzione: si traduca tutto

Operazioni: Si inserisca un nuovo reclamo, aggiornando automaticamente il numero dei reclami per il prodotto coinvolto (se tale numero arriva a 3 il prodotto viene rimosso dal DB, e con esso automaticamente tutte le informazioni collegate)

- 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 NUOVO_RECLAMO
AFTER INSERT ON RECLAMI
REFERENCING NEW AS N
FOR EACH ROW
UPDATE PRODOTTI
SET      NumReclami = NumReclami + 1
WHERE CodProdotto = N.CodProdotto ;
```

```
CREATE OR REPLACE TRIGGER PRODOTTO_RITIRATO
AFTER UPDATE OF NumReclami ON PRODOTTI
REFERENCING NEW AS N
FOR EACH ROW
WHEN ( N.NumReclami >= 3)
DELETE FROM PRODOTTI
WHERE CodProdotto = N.CodProdotto ;
-- La cancellazione da OP e RECLAMI avviene automaticamente mediante la politica di reazione
-- ON DELETE CASCADE specificata per le relative foreign keys
```

```
INSERT INTO RECLAMI VALUES
(:idordine,:codprodotto,:motivo);
```

Sistemi Informativi T

9 gennaio 2025

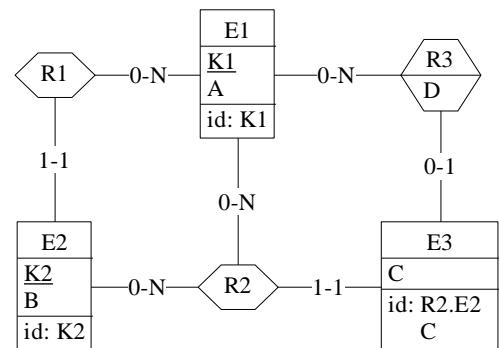
Risoluzione

4) Progettazione logica (6 punti totali)

Dato lo schema concettuale in figura e considerando che:

- a) nessuna associazione viene tradotta separatamente;
- b) istanze di E1 ed E2 associate tramite R1 non possono esserlo anche tramite R2;
- c) due istanze di E3 identificate dalla stessa istanza di E2 non possono entrambe partecipare a R3;

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
);
```

```
CREATE TABLE E3 (
    K2R2   INT NOT NULL REFERENCES E2,
    C       INT NOT NULL,
    K1R2   INT NOT NULL REFERENCES E1,
    K1R3   INT REFERENCES E1,
    D       INT,
    CHECK ((K1R3 IS NULL AND D IS NULL) OR (K1R3 IS NOT NULL AND D IS NOT NULL)),
    PRIMARY KEY (K2R2,C)
);
```

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 b) può essere violato solo inserendo in E3
CREATE OR REPLACE TRIGGER PUNTO_B
BEFORE INSERT ON E3
REFERENCING NEW AS N
FOR EACH ROW
WHEN (EXISTS (SELECT *
 FROM E2
 WHERE N.K2R2 = E2.K2
 AND N.K1R2 = E2.K1R1))
SIGNAL SQLSTATE '70001' ('La tupla inserita in E3 non rispetta il vincolo del punto b)!');

-- Il vincolo al punto c) può essere violato solo inserendo in E3 un'istanza che partecipa a R3
CREATE TRIGGER PUNTO_C
BEFORE INSERT ON E3
REFERENCING NEW AS N
FOR EACH ROW
WHEN (N.K1R3 IS NOT NULL AND
 EXISTS (SELECT *
 FROM E3
 WHERE N.K2R2 = E3.K2R2
 AND E3.K1R3 IS NOT NULL))
SIGNAL SQLSTATE '70002' ('La tupla inserita in E3 non rispetta il vincolo del punto c)!');