

Sistemi Informativi T
15 settembre 2020
Risoluzione

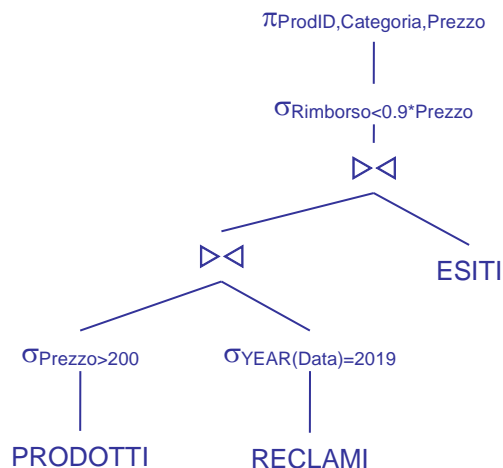
1) Algebra relazionale (3 punti totali):

Date le seguenti relazioni:

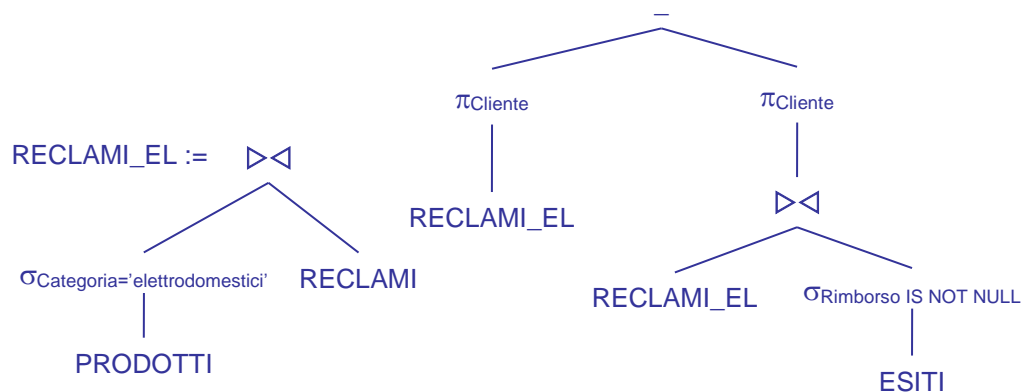
```
PRODOTTI (ProdID, Categoria, Prezzo);
RECLAMI (RID, ProdID, Data, Cliente, Motivo),
  ProdID REFERENCES PRODOTTI;
ESITI (RID, DataSoluzione, Esito, Rimborso*),
  RID REFERENCES RECLAMI;
-- DataSoluzione: la data in cui si è deciso come trattare il reclamo.
-- Esito: descrive se e come il reclamo è stato accolto.
-- NB: se il reclamo non ha ancora un esito il relativo RID non compare
-- in ESITI.
-- 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(8,2).
```

si scrivano in algebra relazionale le seguenti interrogazioni:

- 1.1) [1 p.]** I dati dei prodotti di prezzo maggiore di 200€ che nel 2019 hanno avuto almeno un reclamo per cui il rimborso è stato minore del 90% del prezzo



- 1.2) [2 p.]** I clienti che hanno fatto reclami per prodotti di categoria 'elettrodomestici', ma che non hanno mai avuto un rimborso per tali prodotti



L'operando sinistro della differenza sono tutti i clienti che hanno fatto reclami per prodotti di categoria 'elettrodomestici', l'operando destro quelli di questi che hanno avuto almeno un rimborso per tali reclami

Sistemi Informativi T
15 settembre 2020
Risoluzione

SQL (5 punti totali)

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

- 2.1) [2 p.]** Per ogni categoria di prodotti, il numero medio di giorni trascorsi per avere l'esito di un reclamo respinto (Esito = 'RESPINTO')

```
SELECT    P.Categoria,
          AVG(CAST(DAYS(E.DataSoluzione)-DAYS(R.Data) AS DEC(6,2))
              AS NumMedioGiorni
FROM      PRODOTTI P, RECLAMI R, ESITI E
WHERE     P.ProdID = R.ProdID
AND       R.RID = E.RID
AND       E.Esito = 'RESPINTO'
GROUP BY P.Categoria;
```

-- Il CAST è fondamentale per calcolare correttamente la media

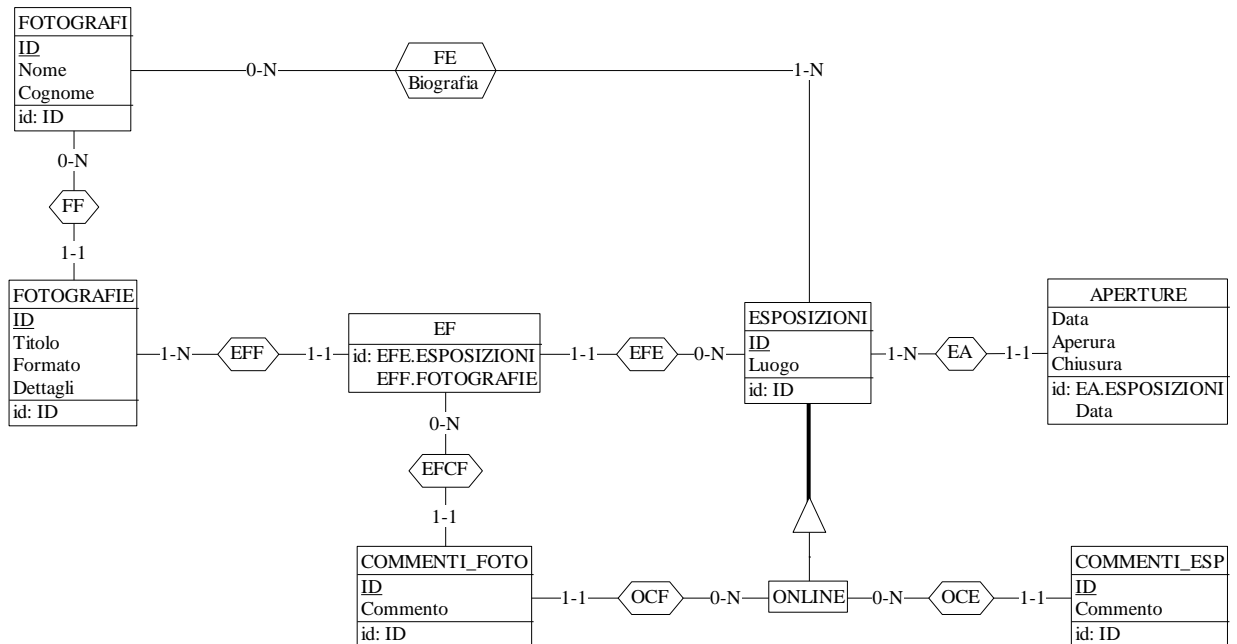
- 2.2) [3 p.]** Per ogni possibile esito, la categoria di prodotti che ha avuto il maggior numero di reclami con quell'esito

```
WITH CAT_ESITI (Categoria, Esito, NumReclami) AS (
    SELECT P.Categoria, E.Esito, COUNT(*)
    FROM   PRODOTTI P, RECLAMI R, ESITI E
    WHERE  P.ProdID = R.ProdID
    AND    R.RID = E.RID
    GROUP BY P.Categoria, E.Esito
)
SELECT CE.Esito, CE.Categoria
FROM   CAT_ESITI CE
WHERE  CE.NumReclami >= ALL ( SELECT CE1.NumReclami
                             FROM   CAT_ESITI CE1
                             WHERE  CE1.Esito = CE.Esito );
```

-- La c.t.e. calcola il numero di reclami per ogni combinazione
-- (Categoria, Esito)

Risoluzione

La cooperativa di fotografi TakeAShot (TAS) cura le esposizioni dei propri iscritti su tutto il territorio nazionale. Di ogni esposizione (che può riguardare anche più fotografi) si tiene traccia del luogo, dei giorni e dei relativi orari di apertura (che possono variare da giorno a giorno), e si fornisce una breve biografia dei fotografi coinvolti (la biografia di un fotografo può variare da un'esposizione all'altra). Per ogni foto esposta TAS mantiene informazioni sul titolo, sul formato e altri dettagli tecnici. Per alcune esposizioni è anche prevista una versione online aperta a tutti, nel qual caso TAS registra i commenti (anonimi), che possono riguardare l'esposizione nel suo complesso o singole fotografie.



- EF è ottenuta per reificazione, per poter associare dei commenti solo a coppie di fotografie ed esposizioni effettivamente presenti, con il vincolo, non esprimibile, che l'esposizione sia (anche) online.
- L'associazione OCF può essere omessa, ma serve a enfatizzare il vincolo sui commenti alle fotografie.
- Anche APERTURE è ottenuta per reificazione. In alternativa si può inserire un attributo composto e ripetuto Aperture in ESPOSIZIONI. Nella soluzione adottata la stessa Data non si può ripetere per una data esposizione.
- Per flessibilità si è posto $\text{min-card}(\text{ESPOSIZIONI}, \text{EFE}) = 0$, in modo da poter inserire nuove esposizioni anche se non sono ancora definite le fotografie che vi compariranno.

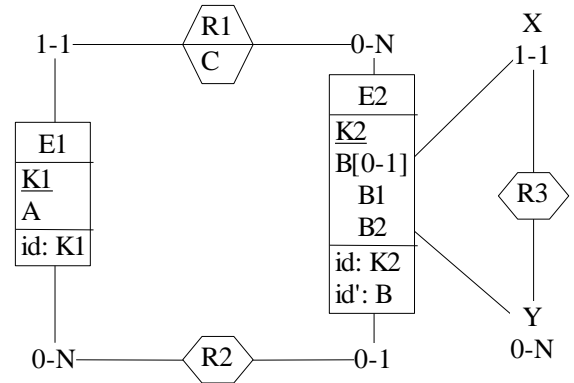
Sistemi Informativi T
15 settembre 2020
Risoluzione

4) Progettazione logica (6 punti totali)

Dato lo schema concettuale in figura e considerando che:

- a) tutti gli attributi sono di tipo INT;
- b) nessuna associazione viene tradotta separatamente;
- c) un'istanza di E2 che partecipa a R2 non può partecipare a R3 con ruolo Y;

4.1) [3 p.] Si progettino gli opportuni schemi relazionali e si definiscano tali schemi in DB2 (sul database SIT_STUD) mediante un file di script denominato **SCHEMI.txt** (o **SCHEMI.sql**)



```
CREATE TABLE E2 (
K2          INT NOT NULL PRIMARY KEY,
B1          INT,
B2          INT,
K2R3Y      INT NOT NULL REFERENCES E2,
K1R2       INT,
CONSTRAINT B CHECK ( (B1 IS NULL AND B2 IS NULL) OR
                      (B1 IS NOT NULL AND B2 IS NOT NULL) ) );
```

```
CREATE TABLE E1 (
K1          INT NOT NULL PRIMARY KEY,
A          INT NOT NULL,
K2R1       INT NOT NULL REFERENCES E2,
C          INT NOT NULL );
```

```
ALTER TABLE E2
ADD CONSTRAINT FKR2 FOREIGN KEY (K1R2) REFERENCES E1;
```

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**, definiti in un file **TRIGGER.txt** (o **TRIGGER.sql**) e usando se necessario il simbolo '@' per terminare gli statement SQL (altrimenti ';')

```
-- Trigger che garantisce l'unicità delle coppie di valori (B1,B2)
CREATE TRIGGER E2_KEY
BEFORE INSERT ON E2
REFERENCING NEW AS N
FOR EACH ROW
WHEN ( EXISTS ( SELECT *
                FROM E2
                WHERE (N.B1,N.B2) = (E2.B1,E2.B2) ) )
SIGNAL SQLSTATE '70001' ('La coppia (B1,B2) non può essere duplicata!');
```

```
-- Trigger che garantisce il rispetto del vincolo al punto c), che può essere violato solo inserendo una tupla in E2
-- che referencia, via R3, un'altra tupla di E2 che partecipa a R2
CREATE TRIGGER PUNTO_C
BEFORE INSERT ON E2
REFERENCING NEW AS N
FOR EACH ROW
WHEN (EXISTS ( SELECT * FROM E2
                WHERE N.K2R3Y = E2.K2
                AND    E2.K1R2 IS NOT NULL ) )
SIGNAL SQLSTATE '70002' ('La tupla inserita referencia una tupla di E2 che partecipa a R2!');
```