

**Sistemi Informativi T**  
**3 febbraio 2021**  
**Risoluzione**

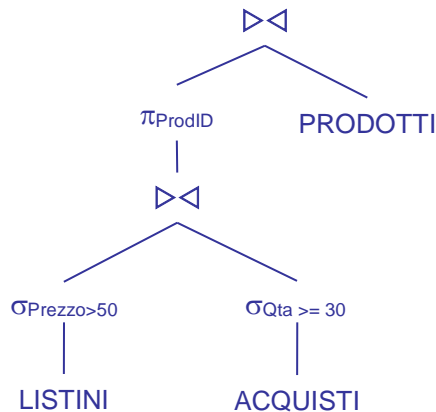
**1) Algebra relazionale (3 punti totali):**

Date le seguenti relazioni:

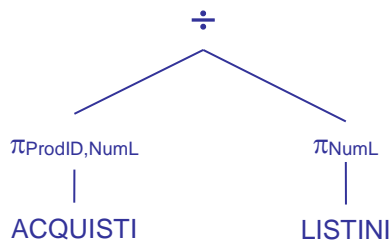
```
PRODOTTI (ProdID, Descrizione);  
LISTINI (NumL, ProdID, Data, Prezzo),  
    ProdID REFERENCES PRODOTTI;  
ACQUISTI (NumL, ProdID, Data, Qta),  
    NumL, ProdID REFERENCES LISTINI;  
-- NumL è un intero progressivo (1,2,...) che indica i successivi listini,  
-- in cui sono riportati i prezzi di tutti i prodotti.  
-- LISTINI.Data dipende solo da NumL e non da ProdID.  
-- I prezzi possono cambiare o meno rispetto al listino precedente.  
-- Per ogni acquisto il prezzo pagato è quello del listino valido  
-- alla Data di acquisto.  
-- Qta è di tipo INT, Prezzo è di tipo DEC(8,2).
```

si esprimano in algebra relazionale le seguenti interrogazioni:

- 1.1) [1 p.]** I dati dei prodotti che hanno avuto almeno una volta un acquisto di 30 o più unità nella stessa data in cui è uscito un nuovo listino, e in quel listino costavano più di 50€



- 1.2) [2 p.]** I codici dei prodotti che sono stati acquistati almeno una volta per ogni listino



**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 prodotto (completo di descrizione) i listini in cui le vendite del prodotto sono state complessivamente più di 5 unità nei primi 60 giorni di validità del listino

```
SELECT  L.ProdID, P.Descrizione, L.NumL
FROM    LISTINI L JOIN PRODOTTI P ON (L.ProdID = P.ProdID)
WHERE   (L.NumL,L.ProdID) IN
      ( SELECT A.NumL,A.ProdID
        FROM   ACQUISTI A
        WHERE  DAYS(A.Data) - DAYS(L.Data) < 60
        GROUP BY A.NumL,A.ProdID
        HAVING SUM(A.Qta) > 5 );
```

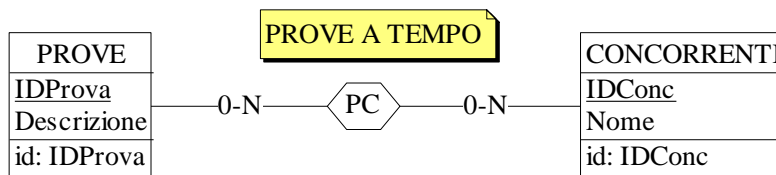
- 2.2) [3 p.]** Considerando solo i prodotti che per un dato listino hanno incassato complessivamente più di 1000€, il listino in cui il numero di tali prodotti è massimo

```
WITH PIUDI1000 (NumL,ProdID) AS (
    SELECT L.NumL, L.ProdID
    FROM   LISTINI L, ACQUISTI A
    WHERE  (L.NumL,L.ProdID) = (A.NumL,A.ProdID)
    GROUP BY L.NumL,L.ProdID
    HAVING SUM(L.Prezzo*A.Qta) > 1000 ),
NUMPRODOTTI (NumL,NumProd) AS (
    SELECT NumL, COUNT(*)
    FROM   PIUDI1000
    GROUP BY NumL )
SELECT *
FROM   NUMPRODOTTI N
WHERE  N.NumProd >= ALL ( SELECT N1.NumProd
                        FROM   NUMPRODOTTI N1 );

-- La prima c.t.e. seleziona i soli prodotti rilevanti, la seconda conta
-- quanti sono per ogni listino
```

**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:

**Specifiche aggiuntive:**

Per ogni concorrente che supera una prova, si tiene traccia del numero di secondi impiegati (NumSecs). Per ogni concorrente si mantengono il numero di prove superate (NumProve, default = 0), il totale dei secondi (TotSecs, default = 0) e la media dei secondi per prova (AvgSecs, default = NULL).

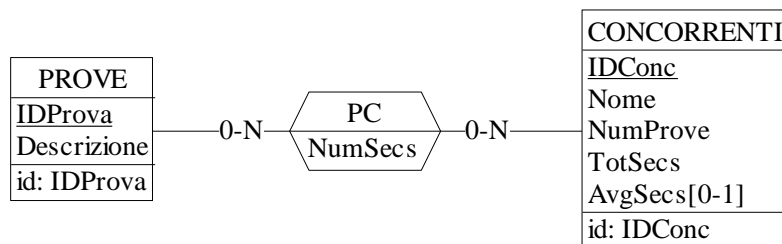
**Traduzione:**

si traduca tutto ad eccezione di PROVE

**Operazioni:**

Si registri il superamento di una prova da parte di un candidato, modificando automaticamente NumProve, TotSecs e AvgSecs.

**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.

```
INSERT INTO PC
```

```
VALUES (:concorrente,:prova,:numsecondi);
```

```
CREATE OR REPLACE TRIGGER UPDATE_DATI
```

```
AFTER INSERT ON PC
```

```
REFERENCING NEW AS N
```

```
FOR EACH ROW
```

```
UPDATE CONCORRENTI
```

```
SET NUMPROVE = NUMPROVE + 1,
```

```
TOTSECS = TOTSECS + N.NUMSECS,
```

```
AVGSECS = DEC((TOTSECS + N.NUMSECS)*1.0/(NUMPROVE + 1),8,2)
```

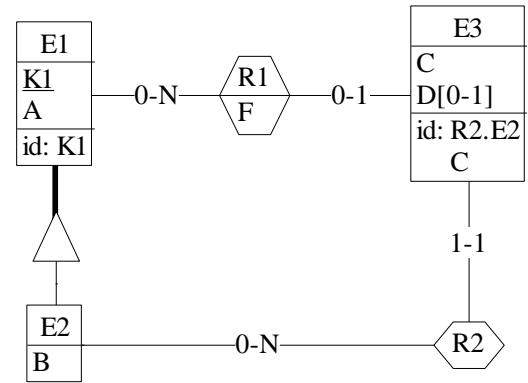
```
WHERE IDCONC = N.IDCONC ;
```

**4) Progettazione logica (6 punti totali)**

Dato lo schema concettuale in figura e considerando che:

- le entità E1 ed E2 vengono tradotte assieme;
- nessuna associazione viene tradotta separatamente;
- le istanze di E3 identificate esternamente dalla stessa istanza di E2 hanno valori di D tutti diversi tra loro;

**4.1) [3 p.]** Si progettino gli opportuni schemi relazionali e si definiscano tali schemi mediante uno script SQL compatibile con DB2



```

CREATE TABLE E1 (
  K1          INT NOT NULL PRIMARY KEY,
  A          INT NOT NULL,           -- il tipo degli attributi non è necessariamente INT
  TIPO2      SMALLINT NOT NULL CHECK (TIPO2 in (1,2)), -- se 2 l'istanza appartiene anche a E2
  B          INT,
  CONSTRAINT E CHECK ((TIPO2 = 1 AND B IS NULL) OR (TIPO2 = 2 AND B IS NOT NULL)) );
  
```

```

CREATE TABLE E3 (
  K1R2       INT NOT NULL REFERENCES E1,
  C          INT NOT NULL,
  D          INT,
  K1R1       INT REFERENCES E1,
  F          INT,
  CONSTRAINT R1 CHECK ((K1R1 IS NULL AND F IS NULL) OR
                       (K1R1 IS NOT NULL AND F IS NOT NULL)),
  PRIMARY KEY (K1R2, 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

```

-- Trigger che garantisce che R2 referenzi un'istanza di E2
CREATE TRIGGER R2_E2
BEFORE INSERT ON E3
REFERENCING NEW AS N
FOR EACH ROW
WHEN ( NOT EXISTS ( SELECT *
                    FROM   E1
                    WHERE N.K1R2 = E1.K1
                    AND    E1.TIPO2 = 2 ) )
SIGNAL SQLSTATE '70001' ('La tupla riferenzia una tupla che non appartiene a E2!');
  
```

```

-- Trigger che garantisce il rispetto del vincolo al punto c)
CREATE TRIGGER PUNTO_C
BEFORE INSERT ON E3
REFERENCING NEW AS N
FOR EACH ROW
WHEN ( EXISTS ( SELECT *
                FROM   E3
                WHERE N.K1R2 = E3.K1R2
                AND    N.D = E3.D ) )
SIGNAL SQLSTATE '70002' ('La tupla inserita non rispetta il vincolo del punto c)! ');
  
```