

**Sistemi Informativi T**  
**31 gennaio 2025**  
**Risoluzione**

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

Date le seguenti relazioni:

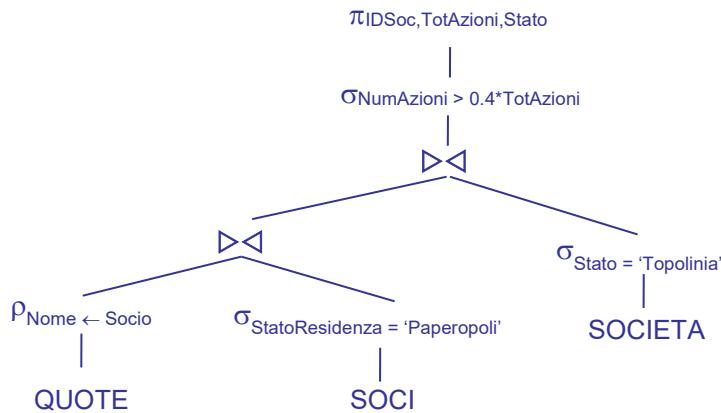
```

SOCIETA (IDSoc, TotAzioni, Stato);
SOCI (Nome, StatoResidenza);
QUOTE (IDSoc, Socio, NumAzioni),
    IDSoc references SOCIETA,
    Socio references SOCI;
-- TotAzioni e NumAzioni sono di tipo INT.
-- Per ogni società, la somma delle azioni possedute dai soci
-- (NumAzioni) è pari al valore di TotAzioni.

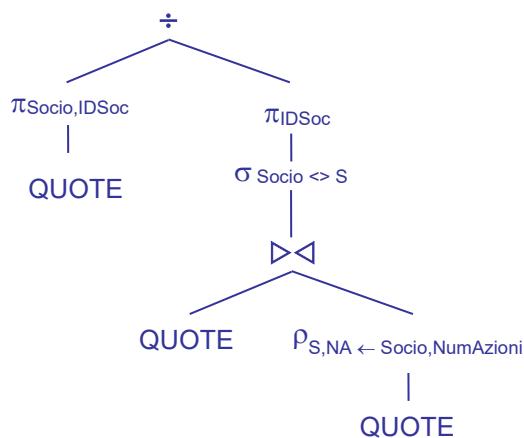
```

si esprimano in algebra relazionale le seguenti interrogazioni:

- 1.1) [1 p.]** I dati delle società di Topolinia in cui almeno un socio di Paperopoli possiede più del 40% delle azioni della società



- 1.2) [2 p.]** I nomi dei soci che possiedono azioni in tutte le società che hanno più di un socio



Il divisore contiene gli identificativi delle società con 2 o più soci.

**Sistemi Informativi T**  
**31 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.]** I dati delle società in cui la quota percentuale di azioni possedute da soci dello stesso stato della società e la quota posseduta da tutti gli altri soci differiscono di meno del 20%

```
SELECT      SA.IDSoc, SA.TotAzioni, SA.Stato
FROM        SOCIETA SA, SOCI S, QUOTE Q
WHERE       SA.IDSoc = Q.IDSoc
AND         S.Nome = Q.Socio
AND         SA.Stato = S.StatoResidenza
GROUP BY    SA.IDSoc, SA.TotAzioni, SA.Stato
HAVING     SUM(Q.NumAzioni)*1.0/SA.TotAzioni > 0.4
AND        SUM(Q.NumAzioni)*1.0/SA.TotAzioni < 0.6 ;
-- Le condizioni nella clausola HAVING sono sufficienti a garantire il
-- rispetto delle specifiche (usare BETWEEN sarebbe impreciso, perché si
-- richiede "meno del 20%", in senso stretto)
```

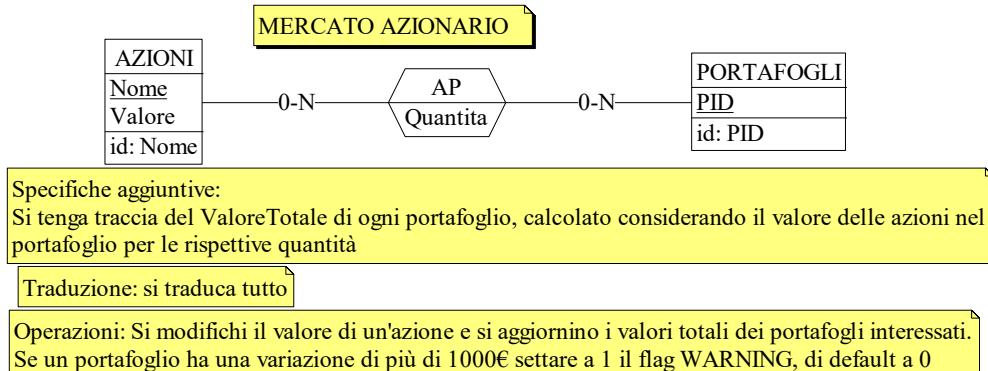
- 2.2) [3 p.]** Le coppie di (nomi di) soci che in tutte le società in comune con almeno 3 soci possiedono almeno il 20% delle azioni a testa

```
SELECT      Q1.Socio AS Socio1, Q2.Socio AS Socio2
FROM        QUOTE Q1, QUOTE Q2, SOCIETA S
WHERE       Q1.IDSoc = Q2.IDSoc
AND         Q1.IDSoc = S.IDSoc
AND         Q1.Socio < Q2.Socio
AND         S.IDSoc IN ( SELECT      Q3.IDSoc
                        FROM        Quote Q3
                        GROUP BY   Q3.IDSoc
                        HAVING     COUNT(*) >= 3 )
GROUP BY    Q1.Socio, Q2.Socio
HAVING     MIN(Q1.NumAzioni*1.0/S.TotAzioni) >= 0.2
AND        MIN(Q2.NumAzioni*1.0/S.TotAzioni) >= 0.2 ;
```

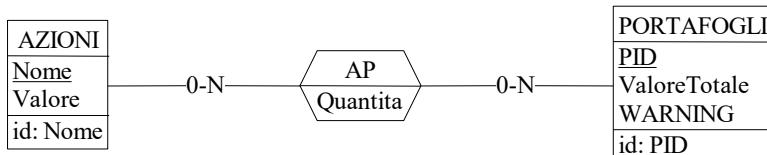
**Sistemi Informativi T**  
**31 gennaio 2025**  
**Risoluzione**

**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:



- 3.1) [1 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) [4 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_VALORE
AFTER UPDATE OF Valore ON AZIONI
REFERENCING NEW AS N OLD AS O
FOR EACH ROW
UPDATE PORTAFOGLI P
SET      P.ValoreTotale = P.ValoreTotale + (N.Valore-O.Valore)*
        ( SELECT AP.Quantita
          FROM   AP
          WHERE  AP.Nome = N.Nome
          AND    AP.PID = P.PID )
WHERE P.PID IN ( SELECT AP.PID
                  FROM   AP
                  WHERE AP.Nome = N.Nome );

```

```

CREATE OR REPLACE TRIGGER SET_FLAG
AFTER UPDATE OF ValoreTotale ON PORTAFOGLI
REFERENCING NEW AS N OLD AS O
FOR EACH ROW
WHEN (ABS(N.ValoreTotale - O.ValoreTotale) > 1000)
UPDATE PORTAFOGLI
SET      WARNING = 1
WHERE  PID = N.PID ;

```

```

UPDATE AZIONI
SET      Valore = Valore + :variazione -- oppure SET Valore = :nuovoValore
WHERE  Nome = :nome ;

```

**Sistemi Informativi T**  
**31 gennaio 2025**  
**Risoluzione**

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

Dato lo schema concettuale in figura e considerando che:

- a) le entità E1 ed E2 vengono tradotte assieme;
- b) l'associazione R1 non viene tradotta separatamente;
- c) le istanze di E1 ed E2 associate a un'istanza di E3 in cui C1 è definito devono avere  $A * B < C1 * C2$  (o  $A * B < C1$  se  $C2$  è nullo);

**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,
    TIPO2   SMALLINT NOT NULL CHECK (TIPO2 IN (1,2)),
    B       INT,
    CHECK ((TIPO2 = 1 AND B IS NULL) OR (TIPO2 = 2 AND B IS NOT NULL)) );
```

```
CREATE TABLE E3 (
    K3      INT NOT NULL PRIMARY KEY,
    C1     INT,
    C2     INT,
    K1E1   INT REFERENCES E1,
    K1E2   INT REFERENCES E1,
    CONSTRAINT R1_DEFINED CHECK ((K1E1 IS NULL AND K1E2 IS NULL) OR
                                  (K1E1 IS NOT NULL AND K1E2 IS NOT NULL)),
    CONSTRAINT C CHECK (C1 IS NOT NULL OR C2 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

-- Trigger che garantisce che K1E2 referenzi un'istanza di E2  
CREATE OR REPLACE TRIGGER R1\_E2  
BEFORE INSERT ON E3  
REFERENCING NEW AS N  
FOR EACH ROW  
WHEN ( EXISTS ( SELECT \* FROM E1  
 WHERE N.K1E2 = E1.K1  
 AND E1.TIPO2 = 1 ) )  
SIGNAL SQLSTATE '70001' ('La tupla inserita referenzia una tupla che non appartiene a E2!');

-- Il vincolo al punto c) può essere violato solo inserendo in E3  
CREATE TRIGGER PUNTO\_C  
BEFORE INSERT ON E3  
REFERENCING NEW AS N  
FOR EACH ROW  
WHEN ( EXISTS ( SELECT \*  
 FROM E1, E1 E2  
 WHERE N.K1E1 = E1.K1  
 AND N.K1E2 = E2.K1  
 AND E1.A \* E2.B >= COALESCE(C2,1) ) )  
SIGNAL SQLSTATE '70002' ('La tupla inserita in E3 non rispetta il vincolo del punto c!)');

-- Se C1 è NULL l'ultima condizione non è soddisfatta e il trigger non segnala errore

