

**Sistemi Informativi T**  
**19 febbraio 2014**  
**Risoluzione**

**Tempo a disposizione: 2:30 ore**

---

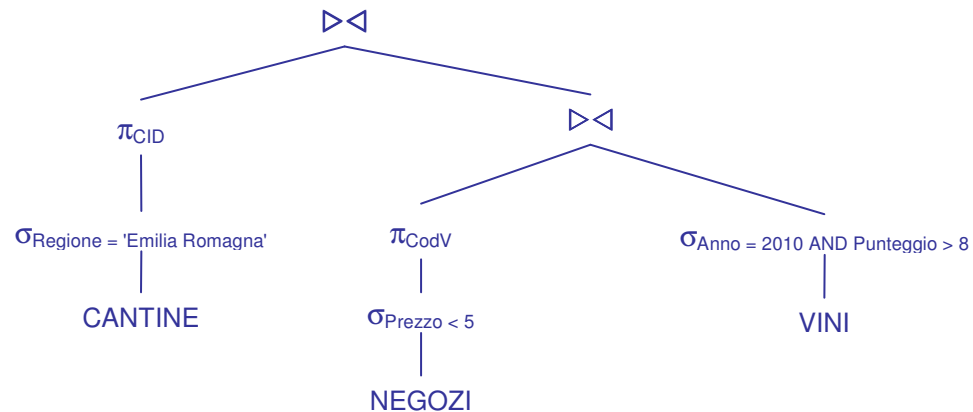
**1) Algebra relazionale (3 punti totali):**

Date le seguenti relazioni:

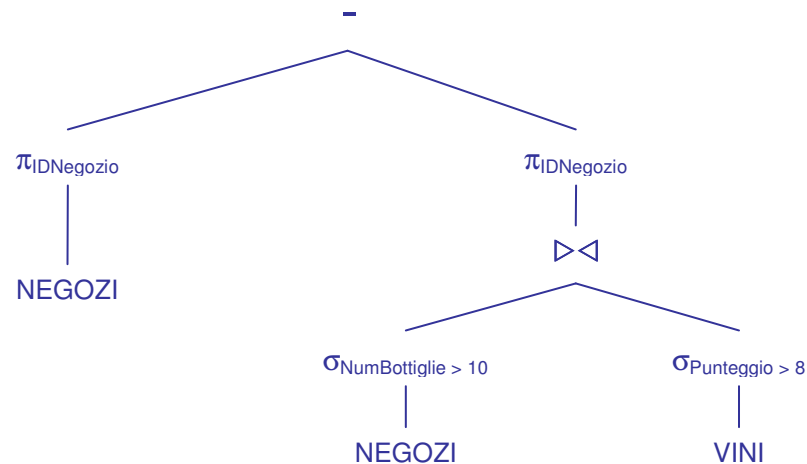
```
CANTINE (CID, Regione) ;  
VINI (CodV, Nome, CID, Anno, Punteggio),  
      CID REFERENCES CANTINE;  
NEGOZI (IDNegozio, CodV, Prezzo, NumBottiglie),  
      CodV REFERENCES VINI;  
--  
-- Prezzo e' di tipo DEC(6,2)  
-- Punteggio e' di tipo DEC(3,1), con valori compresi tra 0 e 10
```

si scrivano in algebra relazionale le seguenti interrogazioni:

- 1.1) [1 p.]** I dettagli dei vini prodotti in Emilia Romagna nel 2010 con un punteggio maggiore di 8 e che si possono acquistare a meno di 5 €



- 1.2) [2 p.]** I negozi che non hanno disponibilità maggiore di 10 bottiglie per nessun vino con punteggio maggiore di 8



**2) SQL (5 punti totali)**

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

- 2.1) [2 p.]** I negozi che non hanno disponibilità maggiore di 10 bottiglie per nessun vino con punteggio maggiore di 8

```
SELECT  N.IDNegozio
FROM    NEGOZI N
      EXCEPT
SELECT  N.IDNegozio
FROM    NEGOZI N JOIN VINI V ON (N.CodV = V.CodV)
WHERE   V.Punteggio > 8
AND     N.NumBottiglie > 10
```

- 2.2) [3 p.]** Per ogni regione con almeno 3 vini con punteggio maggiore di 8, i negozi che ne hanno almeno 10 bottiglie di ognuno

```
WITH
VINI8 (Regione,CodV, IDNegozio, NumBottiglie) AS (
  SELECT C.Regione, V.CodV, N.IDNegozio, N.NumBottiglie
  FROM   CANTINE C, VINI V, NEGOZI N
  WHERE  C.CID = V.CID
  AND    V.CodV = N.CodV
  AND    V.Punteggio > 8
  AND    C.Regione IN ( SELECT C.Regione
                        FROM   CANTINE C, VINI V
                        WHERE  C.CID = V.CID
                        AND    V.Punteggio > 8
                        GROUP BY C.Regione
                        HAVING COUNT(*) >=3 ) )

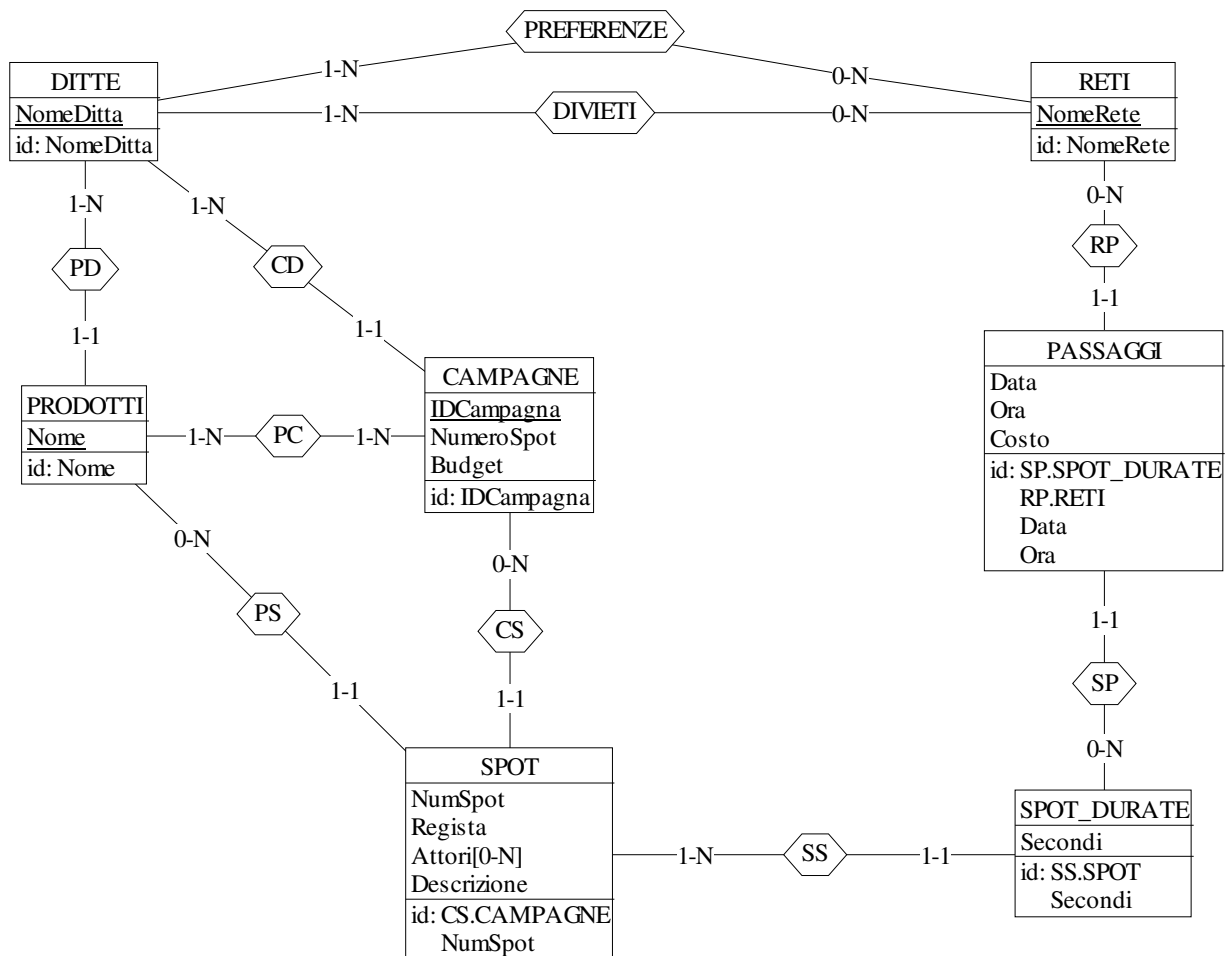
SELECT  V8.Regione, V8.IDNegozio
FROM    VINI8 V8
WHERE   V8.NumBottiglie >= 10
GROUP BY V8.Regione, V8.IDNegozio
HAVING  COUNT(*) = ( SELECT COUNT(*)
                    FROM   VINI8 V2
                    WHERE  V2.Regione = V8.Regione
                    AND    V2.IDNegozio = V8.IDNegozio )

-- La c.t.e. contiene solo dati relativi ai vini con punteggio > 8
-- provenienti da regioni che ne hanno almeno 3; la soluzione confronta,
-- per ogni negozio, se per una data regione il numero di tali vini
-- con almeno 10 bottiglie coincide con il numero totale dei vini
```

**3) Progettazione concettuale (6 punti)**

La società VendiBen (VB) gestisce campagne pubblicitarie su diverse reti televisive. Ogni campagna, relativa a uno o più prodotti di una stessa ditta, si compone di uno o più spot (per esigenze di programmazione e costi, uno stesso spot può essere prodotto in diverse durate di tempo, ad es. 30 e 15 secondi). Ogni spot ha una descrizione, un regista e un cast di attori (in alcuni casi il cast può mancare del tutto), ed è sempre relativo a uno solo dei prodotti della campagna.

Quando una ditta commissiona una campagna pubblicitaria alla VB, vengono stabiliti il numero di spot e il costo complessivo (budget) della campagna. Quando gli spot sono pronti vengono stabiliti tutti i passaggi televisivi, ovvero quando (data e ora) e su quale rete ogni spot verrà trasmesso e in quale durata (il database della VB registra anche il costo di ogni singolo passaggio). Nel predisporre i passaggi, la VB tiene conto del fatto che ogni ditta ha delle preferenze per alcune reti e anche reti su cui non vuole mandare in onda spot dei propri prodotti.

**Commenti:**

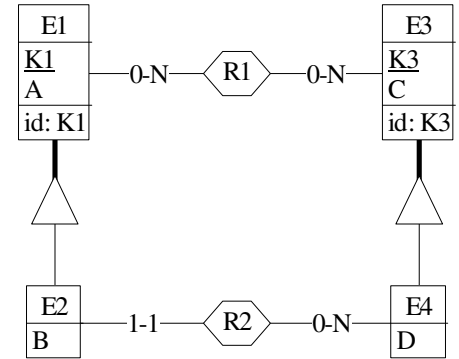
- Non è esprimibile il vincolo che tutti i prodotti di una campagna devono essere della stessa ditta, e che per ogni prodotto oggetto di una campagna deve esistere almeno uno spot della campagna stessa.
- Le cardinalità minima di CAMPAGNE in CS e di PRODOTTI in PS sono pari a 0 perché quando una campagna viene stabilita i relativi spot non sono ancora stati definiti. Per lo stesso motivo è 0 la cardinalità minima di SPOT\_DURATE IN SP.
- Attori e registi potevano anche essere modellati come entità
- L'attributo "Ora" si assume che includa anche i minuti (ad es.: 12:25), altrimenti non è garantita l'identificazione univoca di un passaggio televisivo

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

Dato lo schema concettuale in figura e considerando che:

- a) tutti gli attributi sono di tipo INT;
- b) l'associazione R2 non viene tradotta separatamente;
- c) le entità E1 ed E2 vengono tradotte insieme;
- d) le entità E3 ed E4 vengono tradotte insieme;
- e) un'istanza di E2 non può referenziare tramite R1 un'istanza di E4;

**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**



```
CREATE TABLE E3 (
  K3 INT NOT NULL PRIMARY KEY,
  C INT NOT NULL,
  TIPO34 SMALLINT NOT NULL CHECK (TIPO34 IN (3,4)),    -- 4: istanza anche di E4
  D INT,
  CONSTRAINT E4 CHECK (
    (TIPO34 = 3 AND D IS NULL) OR (TIPO34 = 4 AND D IS NOT NULL) )
);
```

```
CREATE TABLE E1 (
  K1 INT NOT NULL PRIMARY KEY,
  A INT NOT NULL,
  TIPO12 SMALLINT NOT NULL CHECK (TIPO12 IN (1,2)),    -- 2: istanza anche di E2
  B INT,
  K3R2 INT REFERENCES E3,
  CONSTRAINT E2 CHECK (
    (TIPO12 = 1 AND B IS NULL AND K3R2 IS NULL) OR
    (TIPO12 = 2 AND B IS NOT NULL AND K3R2 IS NOT NULL) )
);
```

```
CREATE TABLE R1 (
  K1 INT NOT NULL REFERENCES E1,
  K3 INT NOT NULL REFERENCES E3,
  PRIMARY KEY (K1,K3)
);
```

**4.2) [3 p.]** Per i vincoli non esprimibili a livello di schema si predispongano opportuni **trigger che evitino inserimenti di tuple non corrette**, definiti in un file **TRIGGER.txt** e usando se necessario il simbolo '@' per terminare gli statement SQL (altrimenti ';')

```
CREATE TRIGGER E2_R2_E4
BEFORE INSERT ON E1
REFERENCING NEW AS N
FOR EACH ROW
WHEN (TIPO12 = 2 AND NOT EXISTS ( SELECT * FROM E3
                                   WHERE N.K3R2 = E3.K3
                                   AND E3.TIPO34 = 4 )
)
SIGNAL SQLSTATE '70001' ('La tupla inserita deve referenziare una tupla di E4!');

-- Trigger che garantisce il rispetto del vincolo di cui al punto d)
CREATE TRIGGER PUNTO_D
BEFORE INSERT ON R1
REFERENCING NEW AS N
FOR EACH ROW
WHEN (EXISTS ( SELECT * FROM E1,E3
               WHERE N.K1 = E1.K1
               AND   N.K3 = E3.K3
               AND   E1.TIPO12 = 2
               AND   E3.TIPO34 = 4 ) )
SIGNAL SQLSTATE '70002' ('Le tuple di E2 ed E4 non possono essere associate tramite R1!');
```