

**Sistemi Informativi T**  
**15 giugno 2018**  
**Risoluzione**

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

---

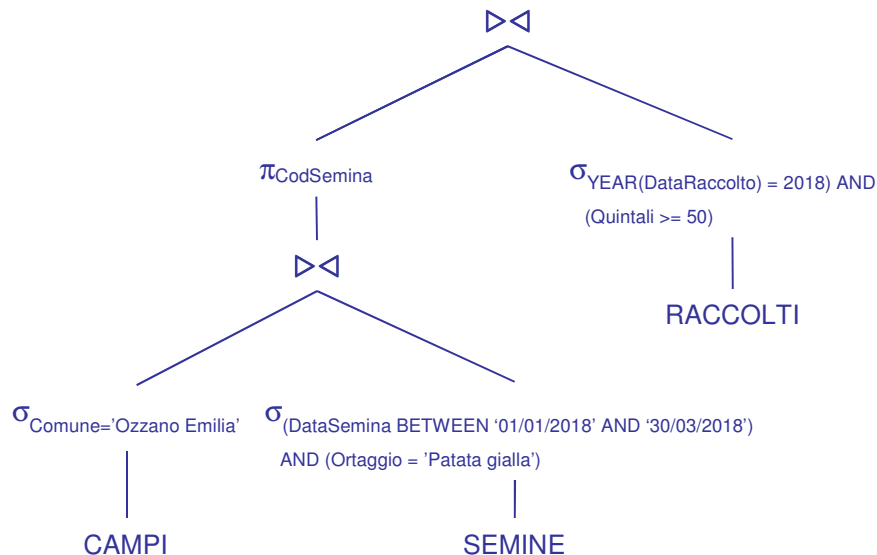
**1) Algebra relazionale (3 punti totali):**

Date le seguenti relazioni:

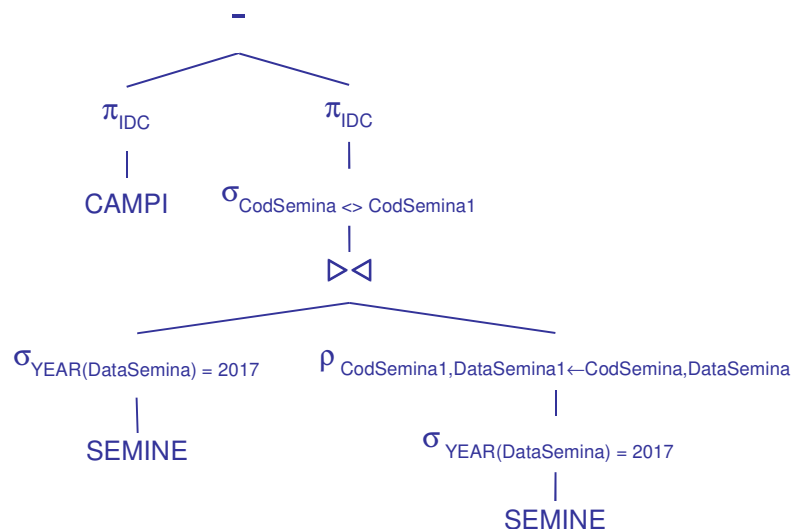
```
CAMPI (IDC, Superficie, Comune);  
SEMIINE (CodSemina, IDC, Ortaggio, DataSemina),  
        IDC references CAMPI;  
RACCOLTI (CodSemina, DataRaccolto, Quintali),  
        CodSemina references SEMIINE;  
-- Superficie è di tipo INT  
-- Quintali è di tipo DEC(6,2)
```

si scrivano in algebra relazionale le seguenti interrogazioni:

- 1.1) [1 p.]** I dati dei raccolti nel comune di Ozzano Emilia del 2018 dell'ortaggio 'Patata gialla' seminati lo stesso anno prima del 31 Marzo e che hanno prodotto almeno 50 quintali



- 1.2) [2 p.]** Gli identificativi dei campi che nel 2017 non hanno avuto due o più semine di uno stesso ortaggio



**Sistemi Informativi T**  
**15 giugno 2018**  
**Risoluzione**

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

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

**2.1) [2 p.]** Gli identificativi dei campi che nel 2017 non hanno avuto tre o più semine di uno stesso ortaggio

```
(SELECT  C.IDC
FROM    CAMPI C)
EXCEPT
(SELECT  S.IDC
FROM    SEMINE S
WHERE   YEAR(S.DataSemina) = 2017
GROUP BY S.IDC, S.Ortaggio
HAVING  COUNT(*) >= 3);
```

**2.2) [3 p.]** Per ogni campo, l'ortaggio che su quel campo ha reso di più, ovvero che ha prodotto in media più quintali considerando i vari raccolti relativi a quel campo

```
WITH
  MEDIE (IDC,Ortaggio,MediaQli) AS (
    SELECT  S.IDC, S.Ortaggio, AVG(R.Quintali)
    FROM    SEMINE S, RACCOLTI R
    WHERE   S.CodSemina = R.CodSemina
    GROUP BY S.IDC, S.Ortaggio
  )

SELECT  M.*
FROM    MEDIE M
WHERE   M.MediaQli = ( SELECT MAX(M1.MediaQli)
                      FROM    MEDIE M1
                      WHERE   M1.IDC = M.IDC );

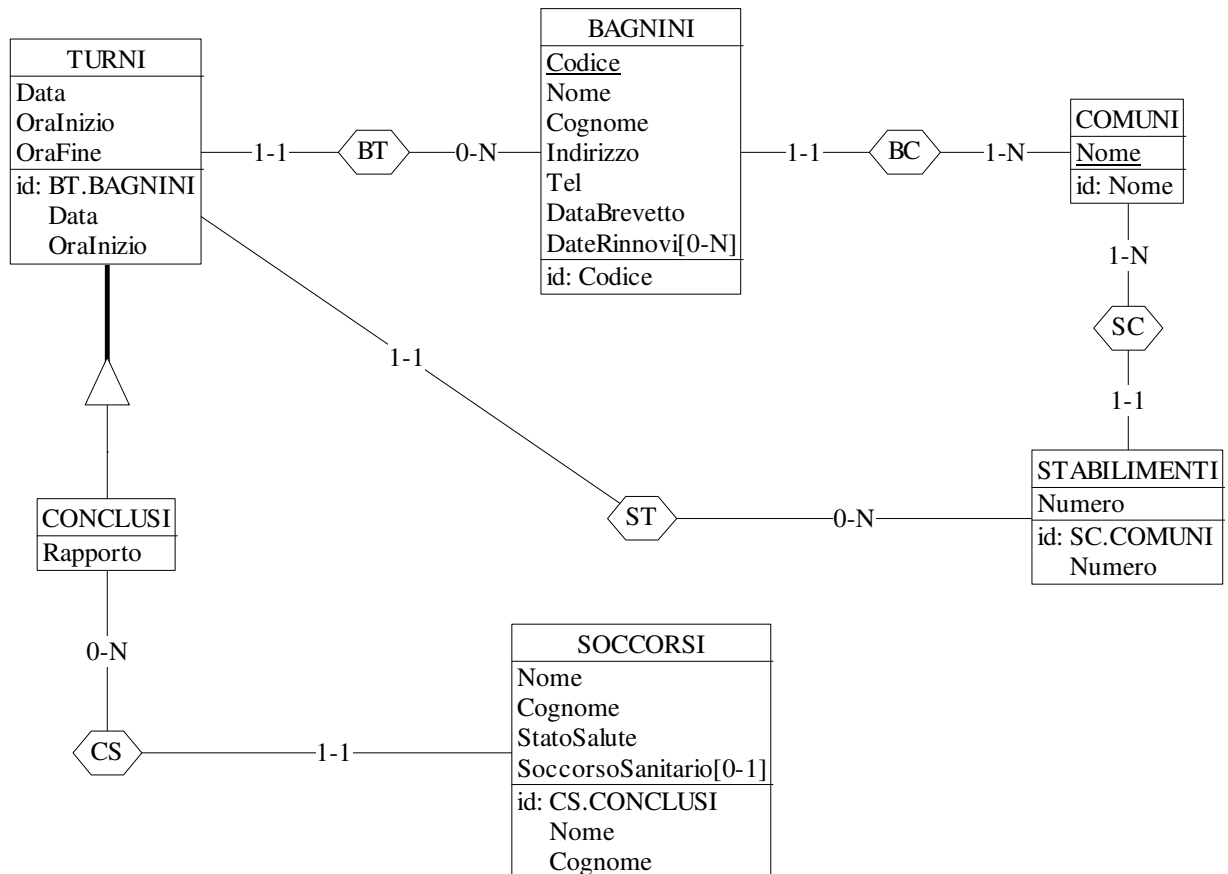
-- La c.t.e. restituisce, per ogni campo e ogni ortaggio, la media dei
-- quintali raccolti
```

3) Progettazione concettuale (6 punti)

La cooperativa di bagnini NuotaSicuro (NS) opera su tutte le spiagge del litorale romagnolo. Di ogni bagnino, identificato da un codice univoco, oltre ai dati anagrafici, si mantiene la data in cui ha acquisito il brevetto, e le date in cui l'ha rinnovato (il rinnovo avviene ogni 2 anni al massimo).

La cooperativa mantiene i turni di tutti i bagnini, ovvero mese, giorno, ora inizio e ora fine servizio. In ogni turno di lavoro, un bagnino svolge la sua attività presso una postazione localizzata di fronte a uno stabilimento balneare (ogni stabilimento ha un numero, univoco all'interno del comune in cui si trova, ad es. Rimini 24). Per motivi organizzativi, ogni bagnino può svolgere le sue attività in più stabilimenti, ma tutti dello stesso comune.

Al termine di ogni turno di lavoro, ogni bagnino ha l'obbligo di compilare un breve rapporto in cui segnalare quanto accaduto durante il turno. Nel caso di persone soccorse in mare, per ognuna di esse va mantenuto un rapporto separato in cui riportare i dati anagrafici della persona, lo stato di salute e l'eventuale ricorso a un soccorso sanitario.



Commenti:

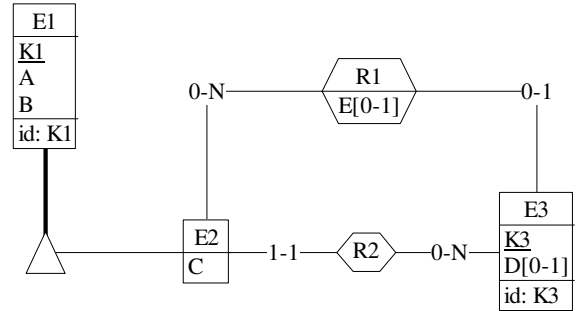
- Il vincolo che un bagnino debba svolgere i suoi turni presso stabilimenti dello stesso comune non è esprimibile in E/R.
- E' possibile definire un'entità PERSONE, che generalizzi BAGNINI e SOCCORSI, anche se in questo caso si rende necessario definire un identificatore condiviso.

**Sistemi Informativi T**  
**15 giugno 2018**  
**Risoluzione**

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

Dato lo schema concettuale in figura e considerando che:

- tutti gli attributi sono di tipo INT;
- le entità E1 ed E2 vengono tradotte insieme;
- le associazioni R1 e R2 non vengono tradotte separatamente;
- un'istanza di E2 è sempre associata, tramite R2, a un'istanza di E3 con D non nullo;



**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,
  D INT,
  K1R1 INT,
  E INT,
  CONSTRAINT R1 CHECK (K1R1 IS NOT NULL OR E IS NULL)
  -- vincolo violato solo se E non è nullo, ma K1R1 lo è
);

CREATE TABLE E12 (
  K1 INT NOT NULL PRIMARY KEY,
  A INT NOT NULL,
  B INT NOT NULL,
  TIPO SMALLINT NOT NULL CHECK (TIPO IN (1,2)),      -- 2: istanza anche di E2
  K3R2 INT REFERENCES E3,
  C INT,
  CONSTRAINT E2 CHECK ((TIPO = 1 AND C IS NULL AND K3R2 IS NULL) OR
    (TIPO = 2 AND C IS NOT NULL AND K3R2 IS NOT NULL))
);

ALTER TABLE E3
ADD CONSTRAINT FK_E2_R1 FOREIGN KEY (K1R1) REFERENCES E12 ;
```

**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 ';')

```
-- Trigger che garantisce che una tupla di E3 referenzi, tramite K1R1, una tupla di E2
CREATE TRIGGER R1_E2
BEFORE INSERT ON E3
REFERENCING NEW AS N
FOR EACH ROW
WHEN (NOT EXISTS ( SELECT * FROM E12
                   WHERE   N.K1R1 = E12.K1
                   AND     E12.TIPO = 2 ) )
SIGNAL SQLSTATE '70001' ('La tupla inserita deve referenziare una tupla di E2!');

-- Trigger che garantisce il rispetto del vincolo al punto d)
CREATE TRIGGER PUNTO_D
BEFORE INSERT ON E12
REFERENCING NEW AS N
FOR EACH ROW
WHEN ( EXISTS ( SELECT *
                FROM   E3
                WHERE  N.K3R2 = E3.K3
                AND    E3.D IS NULL ) )
SIGNAL SQLSTATE '70002' ('La tupla è associata a una tupla di E3 con D nullo!');
```