

**Sistemi Informativi T**  
**15 settembre 2021**  
**Risoluzione**

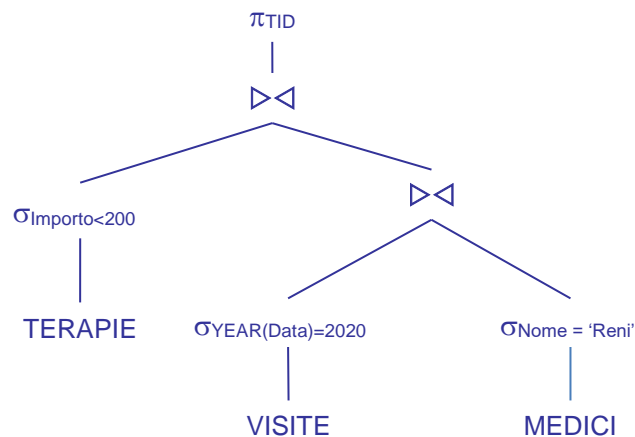
**1) Algebra relazionale (3 punti totali):**

Date le seguenti relazioni:

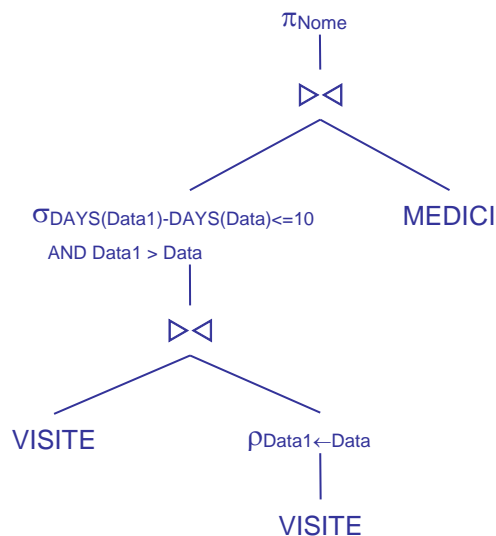
```
MEDICI (MID, Nome) ;  
TERAPIE (TID, Paziente, Importo) ;  
VISITE (TID, Data, MID) ,  
      TID REFERENCES TERAPIE,  
      MID REFERENCES MEDICI ;  
--  
-- Importo è di tipo DEC(8,2)  
-- In generale, una terapia prevede più visite di controllo,  
-- che possono essere eseguite da uno stesso medico o anche da medici diversi
```

si esprimano in algebra relazionale le seguenti interrogazioni:

- 1.1) [1 p.]** Gli identificativi delle terapie di importo inferiore a 200€ in cui almeno una visita è stata fatta dal medico Reni nel 2020



- 1.2) [2 p.]** I nomi dei medici che hanno fatto almeno 2 visite relative alla stessa terapia a distanza di 10 giorni o meno



**Sistemi Informativi T**  
**15 settembre 2021**  
**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 nomi dei medici che nel 2021 hanno eseguito almeno 3 visite per una stessa terapia di importo maggiore di 500€

```
SELECT  M.Nome
FROM    MEDICI M
WHERE   M.MID IN
        ( SELECT      V.MID
          FROM        TERAPIE T, VISITE V
          WHERE       T.TID = V.TID
          AND         T.Importo > 500
          AND         YEAR(V.Data) = 2021
          GROUP BY   V.MID, V.TID
          HAVING      COUNT(*) >= 3  )  ;
```

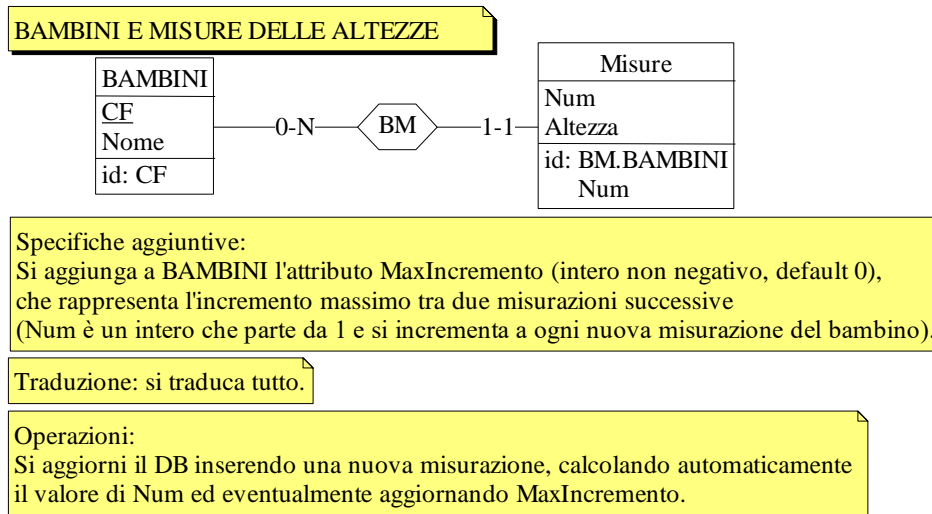
- 2.2) [3 p.]** I dati della terapia di importo massimo, considerando solo le terapie in cui tutte le visite (almeno 2) sono state con uno stesso medico

```
WITH TERAPIE_UN_MEDICO(TID) AS (
  SELECT V.TID
  FROM   VISITE V
  GROUP BY V.TID
  HAVING COUNT(*) > 1 AND COUNT(DISTINCT V.MID) = 1 )
SELECT  T.*
FROM    TERAPIE T, TERAPIE_UN_MEDICO TUM
WHERE   T.TID = TUM.TID
AND     T.Importo = ( SELECT MAX(T1.Importo)
                    FROM    TERAPIE T1, TERAPIE_UN_MEDICO TUM1
                    WHERE   T1.TID = TUM1.TID ) ;

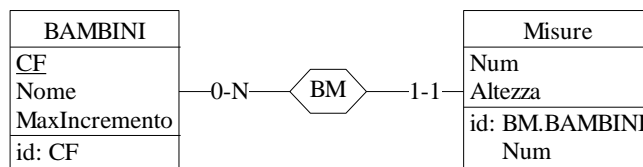
-- La c.t.e. selezione le sole terapie che rispettano i requisiti
-- (almeno 2 visite e tutte con lo stesso medico)
```

**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 produca uno schema ESE3-modificato secondo le Specifiche aggiuntive;



**3.2) [1 p.]** 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 INCREMENTA_NUM
BEFORE INSERT ON Misure
REFERENCING NEW AS NewM
FOR EACH ROW
SET NewM.Num = 1 + COALESCE((SELECT Max(Num) FROM Misure WHERE NewM.CF = CF),0);
-- gestisce anche il caso della prima misurazione, in quanto COALESCE(NULL,0) = 0
  
```

```

CREATE OR REPLACE TRIGGER AGGIORNA_MAX_INCREMENTO
AFTER INSERT ON Misure
REFERENCING NEW AS NewM
FOR EACH ROW
WHEN (NewM.Num >= 2) -- almeno 2 misure
UPDATE BAMBINI
SET MaxIncremento = Max(MaxIncremento,
    ( SELECT NewM.Altezza-M.Altezza
      FROM   Misure M
      WHERE  NewM.CF = M.CF
      AND    M.Num = NewM.Num-1 ) )
WHERE CF = NewM.CF;
  
```

```

INSERT INTO BAMBINI VALUES ('ABCDE','Luca',DEFAULT);
  
```

```

INSERT INTO MISURE(CF,Altezza)
VALUES('ABCDE',86);
  
```

```

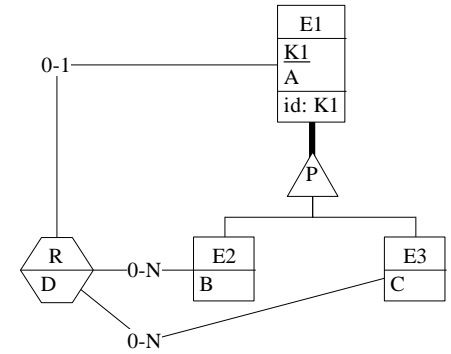
INSERT INTO MISURE(CF,Altezza)
VALUES('ABCDE',91);
  
```

**Sistemi Informativi T**  
**15 settembre 2021**  
**Risoluzione**

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

Dato lo schema concettuale in figura e considerando che:

- le entità E1 ed E2 vengono tradotte assieme, e separatamente da E3
- l'associazione R non viene tradotta separatamente;
- ogni coppia di istanze di E2 ed E3 che partecipa a R dai rami 0-N non può avere  $C > B$ .



**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,
  TIPO23     SMALLINT NOT NULL CHECK (TIPO23 IN (2,3)), -- 2: appartiene a E2, 3: appartiene a E3
  B          INT,
  K1E2       INT REFERENCES E1,
  K1E3       INT,
  D          INT,
  CONSTRAINT E2E3 CHECK ( (TIPO23 = 2 AND B IS NOT NULL) OR (TIPO23 = 3 AND B IS NULL)),
  CONSTRAINT R CHECK ( (K1E2 IS NOT NULL AND K1E3 IS NOT NULL AND D IS NOT NULL) OR
    (K1E2 IS NULL AND K1E3 IS NULL AND D IS NULL)) );
```

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

```
ALTER TABLE E1
ADD CONSTRAINT FK_E3 FOREIGN KEY (K1E3) REFERENCES E3 ;
```

**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
-- (funziona anche se la tupla inserita non partecipa a R, ovvero se K1E2 è nullo)
CREATE OR REPLACE TRIGGER R_E2
BEFORE INSERT ON E1
REFERENCING NEW AS N
FOR EACH ROW
WHEN ( EXISTS ( SELECT *
                FROM   E1
                WHERE  N.K1E2 = E1.K1
                AND    E1.TIPO23 = 3 ) )
SIGNAL SQLSTATE '70001' ('La tupla referencia una tupla che non appartiene a E2!');
```

```
-- Trigger che garantisce il rispetto del vincolo al punto c)
CREATE OR REPLACE TRIGGER PUNTO_C
BEFORE INSERT ON E1
REFERENCING NEW AS N
FOR EACH ROW
WHEN ( EXISTS ( SELECT * FROM E1, E3
                WHERE  N.K1E2 = E1.K1
                AND    N.K1E3 = E3.K1
                AND    E1.B < E3.C ) )
SIGNAL SQLSTATE '70002' ('La tupla inserita non rispetta il vincolo del punto c! ');
```

-- L'inserimento di una tupla in E1 con TIPO23 =3 deve prevedere l'inserimento anche in E3 nella stessa transazione