

**Sistemi Informativi T**  
**17 febbraio 2010**  
**Risoluzione**

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

---

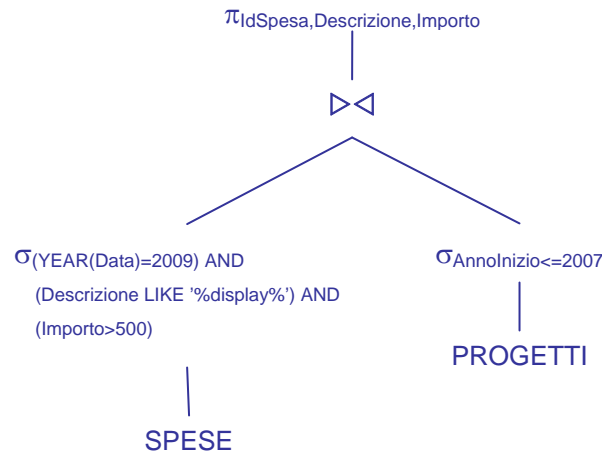
**1) Algebra relazionale (3 punti totali):**

Date le seguenti relazioni:

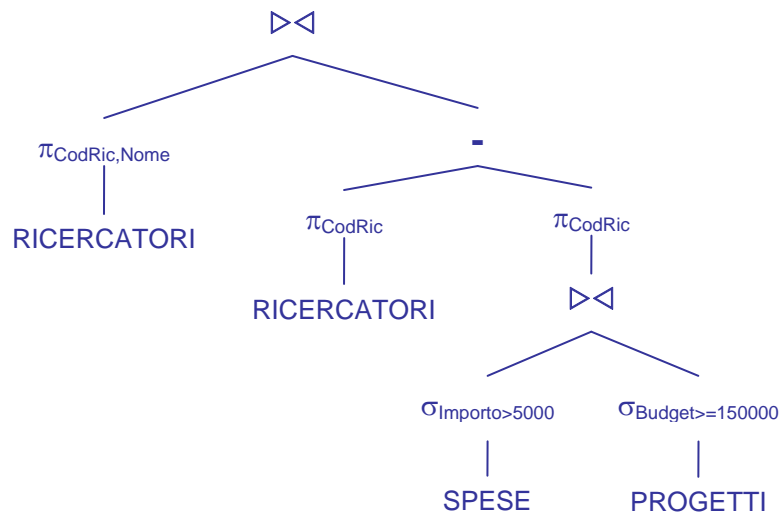
```
PROGETTI(CodProg, Budget, AnnoInizio, AnnoFine);  
RICERCATORI(CodRic, Nome, DataNascita);  
SPESE(IdSpesa, Descrizione, Data, Importo, CodProg, CodRic),  
CodProg references PROGETTI, CodRic references RICERCATORI;  
-- ogni spesa grava sul budget di un progetto  
-- ed e' imputata a un ricercatore  
-- Importo è di tipo DECIMAL(7,2)
```

si scrivano in algebra relazionale le seguenti interrogazioni:

- 1.1) [1 p.]** Identificatore, descrizione e importo delle spese eseguite nel 2009 relative a progetti iniziati nel 2007 o prima, e in cui la descrizione contiene il termine “display” e l’importo è superiore a 500 €



- 1.2) [2 p.]** Codice e nome di tutti i ricercatori che non hanno mai eseguito una spesa di importo superiore ai 5000 € relativa a un progetto con un budget di almeno 150000 €



**Sistemi Informativi T**  
**17 febbraio 2010**  
**Risoluzione**

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

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

**2.1) [2 p.]** Per ogni progetto di durata almeno triennale, l'importo complessivo delle spese sostenute, escludendo quelle di importo inferiore ai 100 € e visualizzando il risultato solo se il totale così ottenuto è almeno pari al 50% del budget del progetto

```
SELECT P.CodProg, SUM(S.Importo) AS TotSpese
FROM PROGETTI P, SPESE S
WHERE P.CodProg = S.CodProg
AND P.AnnoFine - P.AnnoInizio >= 3
AND S.Importo >= 100
GROUP BY P.CodProg, P.Budget
HAVING SUM(S.Importo) >= 0.5*P.Budget
-- e' necessario raggruppare anche su P.Budget per poter scrivere la
-- condizione nella clausola HAVING
```

**2.2) [3 p.]** Per ogni fascia di età dei ricercatori (20-29, 30-39, ecc.), la media delle spese (in formato DECIMAL(7,2)) sostenute dai ricercatori nella fascia e relative a progetti in cui vi sono state almeno 100 spese distinte

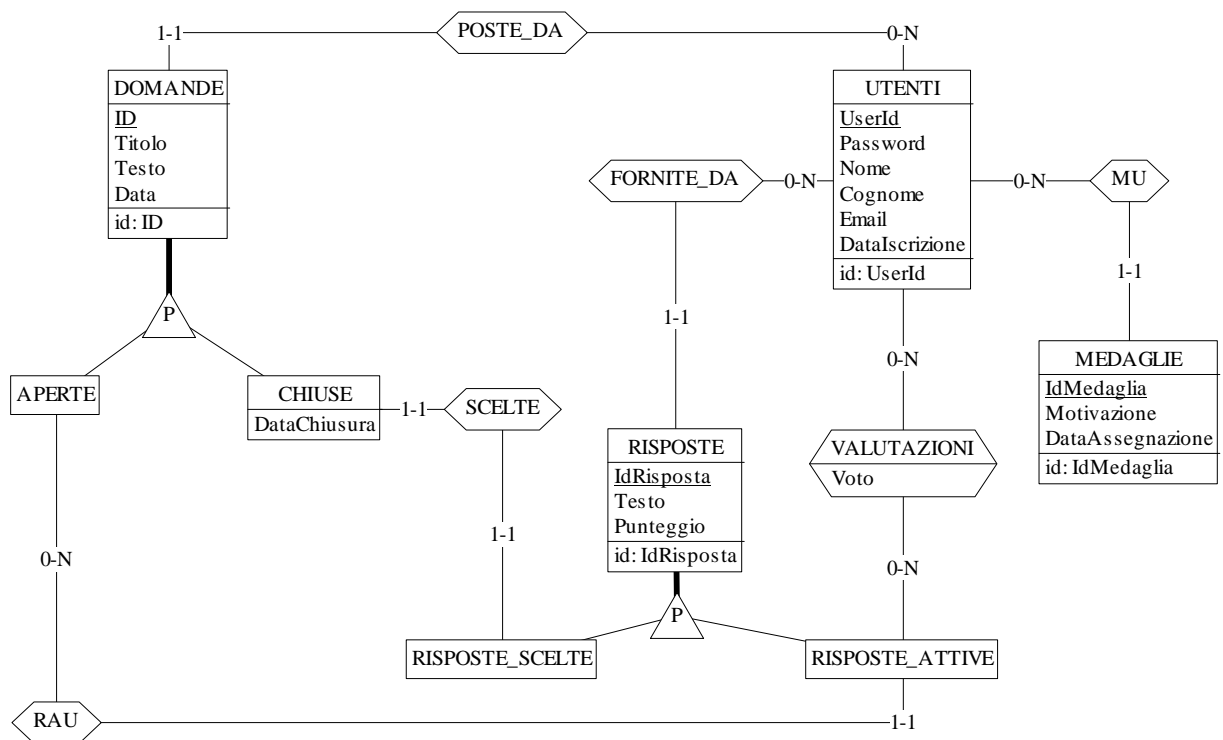
```
SELECT (YEAR(CURRENT DATE - R.DataNascita)/10)*10 AS FasciaEta,
       CAST(AVG(S.Importo) AS DECIMAL(7,2)) AS SpesaMedia
FROM RICERCATORI R, SPESE S
WHERE R.CodRic = S.CodRic
AND S.CodProg IN ( SELECT S1.CodProg
                   FROM Spese S1
                   GROUP BY S1.CodProg
                   HAVING COUNT(*) >= 100 )
GROUP BY YEAR(CURRENT DATE - R.DataNascita)/10
-- si ricorda che non e' possibile eseguire il GROUP BY su FasciaEta
```

**Sistemi Informativi T**  
**17 febbraio 2010**  
**Risoluzione**

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

Il sito NotNullPrimaryKey.com (NNPK) organizza le domande e le risposte di una comunità di amministratori di DBMS. Ogni utente è identificato univocamente dal suo userid, e ne vengono memorizzati anche nome, cognome, indirizzo email, password per l'accesso al sito e data di iscrizione. Ogni utente può porre una domanda, della quale si memorizza il titolo, il testo e la data. Appena pubblicata sul sito, la domanda viene considerata "aperta" e tutti gli utenti possono dare una risposta. Ogni risposta ha un punteggio, calcolato come la media dei voti dati dagli altri utenti alla risposta, che così facendo ne valutano la bontà. Uno stesso utente può fornire più risposte alla stessa domanda. Quando l'utente che ha posto la domanda si ritiene soddisfatto delle risposte ricevute può far passare la domanda nello stato di "chiuso". Di una domanda chiusa si memorizza la data di chiusura e una risposta scelta dall'utente con il relativo punteggio medio, eliminando quindi dal sistema tutti gli altri dettagli.

Per incentivare l'interazione con il sito, NNPK assegna agli utenti delle medaglie virtuali. Ogni medaglia è descritta da una motivazione e dalla data di assegnazione.

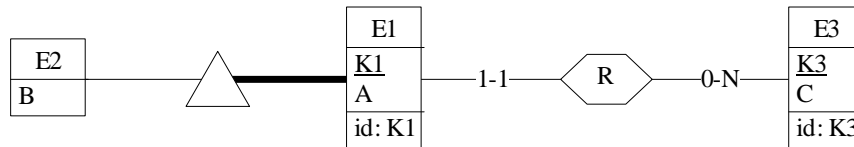


**Commenti**

- Le due gerarchie radicate nelle entità **DOMANDE** e **RISPOSTE** sono praticamente forzate, date le diverse proprietà (attributi e/o associazioni) che caratterizzano le rispettive entità figlie.

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

Dato lo schema concettuale in figura



e considerando che:

- tutti gli attributi sono di tipo INT;
- l'associazione R non viene tradotta separatamente;
- le entità E1 ed E2 vengono tradotte assieme;
- vale la dipendenza funzionale  $B \rightarrow C$ , ovvero diverse istanze di E2 con lo stesso valore di B sono associate a istanze di E3 con lo stesso valore di C;

**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 );

CREATE TABLE E12(
  K1 INT NOT NULL PRIMARY KEY,
  A INT NOT NULL,
  SEL2 SMALLINT CHECK (SEL2 IN (1,2)), -- SEL2 = 2 : la tupla rappresenta anche un'istanza di E2
  B INT,
  K3 INT NOT NULL REFERENCES E3,
  CONSTRAINT E2_SUBSET CHECK ((SEL2 = 2 AND B IS NOT NULL)
                               OR (SEL2 = 1 AND B IS NULL)) );
  
```

**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 il simbolo '@' per terminare gli statement SQL

```

CREATE TRIGGER INS_E12
NO CASCADE BEFORE INSERT ON E12
REFERENCING NEW AS N
FOR EACH ROW MODE DB2SQL
WHEN ( EXISTS ( SELECT *
                FROM E12, E3 E3A, E3 E3B
                WHERE E12.B = N.B
                  AND E12.K3 = E3A.K3
                  AND N.K3 = E3B.K3
                  AND E3A.C <> E3B.C ))
SIGNAL SQLSTATE '70001' ('Dipendenza funzionale B --> C violata!')@
-- SPIEGAZIONE:
-- la clausola WHEN seleziona innanzitutto da E12 le tuple che hanno un valore di B uguale a quello
-- inserito (E12.B = N.B); per tali tuple si esegue il join con E3 (alias E3A), che individua quindi dei
-- valori di E3A.C (in realtà uno solo, poiché per ipotesi il DB è in uno stato consistente);
-- la condizione N.K3 = E3B.K3 individua una tupla in E3 (alias E3B) e l'ultima condizione infine
-- verifica che il valore di C presente in tale tupla sia diverso da quello individuato precedentemente;
-- se così è si ha una violazione della dipendenza funzionale
  
```