

Tempo a disposizione: 2:30 ore

1) Algebra relazionale (3 punti totali):

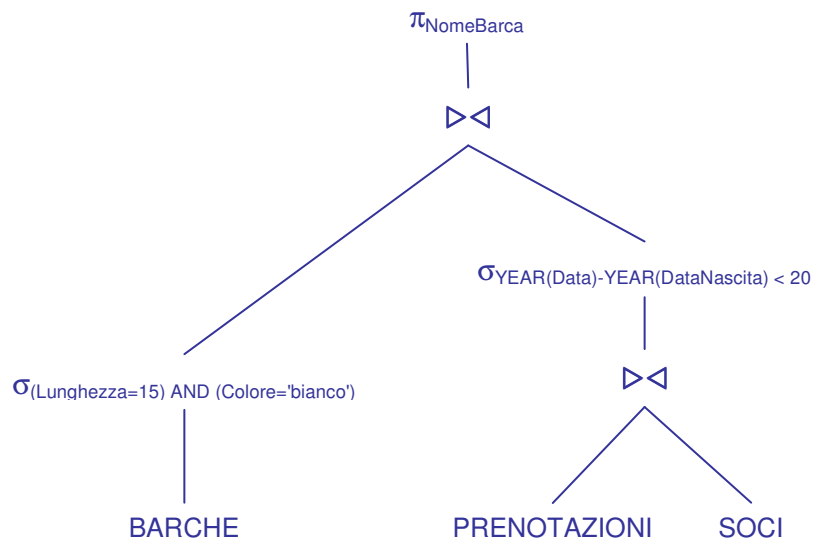
Date le seguenti relazioni:

```

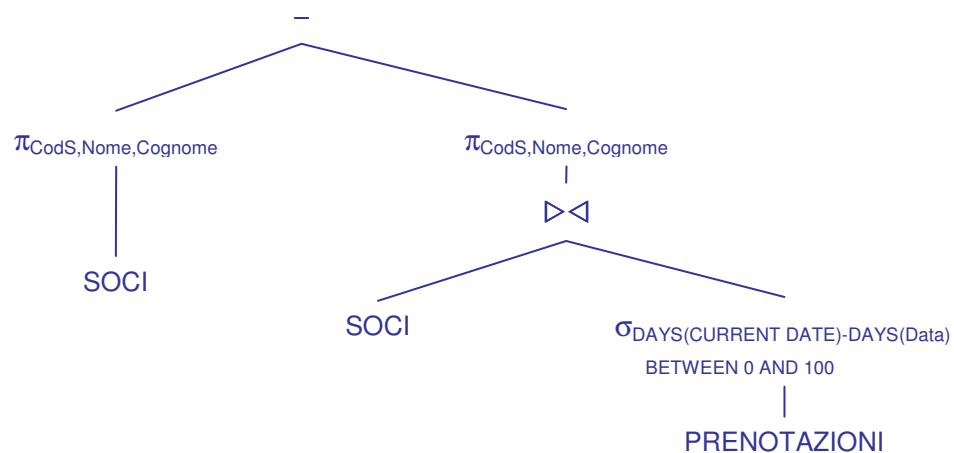
BARCHE (NomeBarca, Colore, Lunghezza);
-- Lunghezza e' in formato Dec(4,2)
SOCI (CodS, Nome, Cognome, DataNascita);
PRENOTAZIONI (NomeBarca, CodS, Data),
NomeBarca REFERENCES BARCHE, CodS REFERENCES SOCI;
-- Data si riferisce alla data in cui si vuole usare una barca,
-- NON alla data in cui e' stata eseguita la prenotazione

```

- 1.1) [1 p.]** Le barche di 15 metri e colore bianco usate da soci con meno di 20 anni all'atto della prenotazione (si ignorino per semplicità i giorni della nascita e della prenotazione, e si considerino solo gli anni delle date)



- 1.2) [2 p.]** Codice, nome e cognome dei soci che non hanno usato nessuna barca negli ultimi 100 giorni



Si vedano i commenti alla query SQL 2.1

Sistemi Informativi T
17 gennaio 2012
Risoluzione

2) SQL (5 punti totali)

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

2.1) [2 p.] Codice, nome e cognome dei soci che non hanno usato nessuna barca negli ultimi 100 giorni

```
SELECT    S.CodS, S.Nome, S.Cognome
FROM      SOCI S
WHERE     S.CodS NOT IN ( SELECT P.CodS
                        FROM PRENOTAZIONI P
                        WHERE DAYS(CURRENT DATE)-DAYS(P.Data)
                              BETWEEN 0 AND 100 )

-- Non si puo' usare solo il predicato DAYS(CURRENT DATE) - DAYS(Data) <= 100
-- nella subquery perche' questo e' sempre soddisfatto per prenotazioni in
-- date future (DAYS(CURRENT DATE)-DAYS(P.Data) < 0), che vanno viceversa
-- escluse
```

2.2) [3 p.] La lunghezza di barca che è stata prenotata complessivamente il maggior numero di volte

```
WITH
TOTPRENOTAZIONI(Lunghezza, Numero) AS (
    SELECT B.Lunghezza, COUNT(*)
    FROM   PRENOTAZIONI P, BARCHE B
    WHERE  P.NomeBarca = B.NomeBarca
    GROUP BY B.Lunghezza )
SELECT T.Lunghezza
FROM   TOTPRENOTAZIONI T
WHERE  T.Numero = ( SELECT MAX(T1.Numero)
                  FROM TOTPRENOTAZIONI T1 )

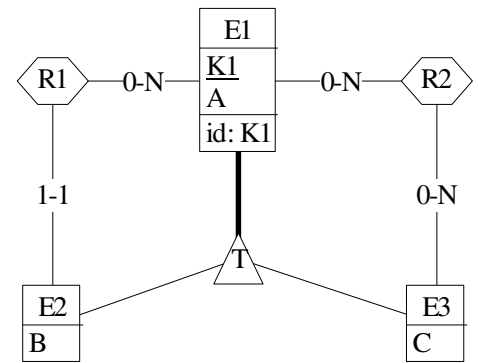
-- Nella Common Table Expression si contano le prenotazioni per ogni valore
-- di lunghezza di barca
```


Progettazione logica (6 punti totali)

Dato lo schema concettuale in figura e considerando che:

- tutti gli attributi sono di tipo INT;
- l'associazione R1 non viene tradotta separatamente;
- le entità E1, E2 ed E3 vengono tradotte assieme;
- un'istanza di E2 non è mai associata, tramite R1, a istanze di E3 con $C < 20$;

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 E1 (
  K1 INT NOT NULL PRIMARY KEY,
  A INT NOT NULL,
  TIPO2 SMALLINT NOT NULL CHECK (TIPO2 IN (0,1)),      -- 1: istanza anche di E2
  TIPO3 SMALLINT NOT NULL CHECK (TIPO3 IN (0,1)),      -- 1: istanza anche di E3
  B INT,
  K1R1 INT REFERENCES E1,
  C INT,
  CONSTRAINT E2 CHECK
    ( (TIPO2 = 1 AND B IS NOT NULL AND K1R1 IS NOT NULL) OR
      (TIPO2 = 0 AND B IS NULL AND K1R1 IS NULL) ),
  CONSTRAINT E3 CHECK ( (TIPO3 = 1 AND C IS NOT NULL) OR (TIPO3 = 0 AND C IS NULL) ),
  CONSTRAINT GERARCHIA_TOTALE CHECK ( TIPO2 = 1 OR TIPO3 = 1) );
```

```
CREATE TABLE R2 (
  K1E1 INT NOT NULL REFERENCES E1,
  K1E3 INT NOT NULL REFERENCES E1,
  PRIMARY KEY (K1E1,K1E3) );
```

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

```
-- Quando si inserisce una tupla in R2 bisogna verificare che il valore di K1E3 referenzi un'istanza di E3
CREATE TRIGGER R2_REFERENZIA_E3
NO CASCADE BEFORE INSERT ON R2
REFERENCING NEW AS N
FOR EACH ROW
WHEN (NOT EXISTS ( SELECT *
                   FROM E1
                   WHERE E1.K1 = N.K1E3
                   AND E1.TIPO3 = 1) )
SIGNAL SQLSTATE '70001' ('La tupla inserita deve referenziare una istanza di E3!')@
```

```
-- Per garantire il rispetto del vincolo di cui al punto d) è necessario impostare il seguente trigger:
CREATE TRIGGER PUNTO_D
NO CASCADE BEFORE INSERT ON R2
REFERENCING NEW AS N
FOR EACH ROW
WHEN (EXISTS (SELECT *
              FROM E1
              WHERE E1.K1 = N.K1R1    -- K1R1 e' definito solo per istanze di E2
              AND E2.C < 20 ))        -- C e' definito solo per istanze di E3
SIGNAL SQLSTATE '70001' ('La tupla inserita referencia una tupla con C<20!')@
-- Non serve verificare i tipi delle istanze inserite e referenziate perche' garantiti dai check E2 ed E3 in E1
```