

Tempo a disposizione: 2:30 ore

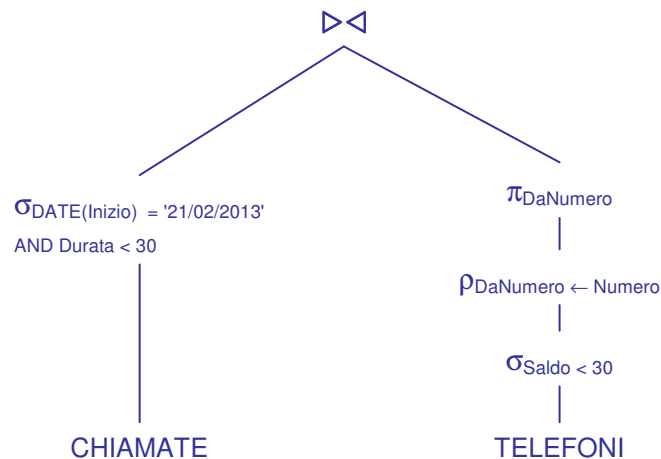
1) Algebra relazionale (3 punti totali):

Date le seguenti relazioni:

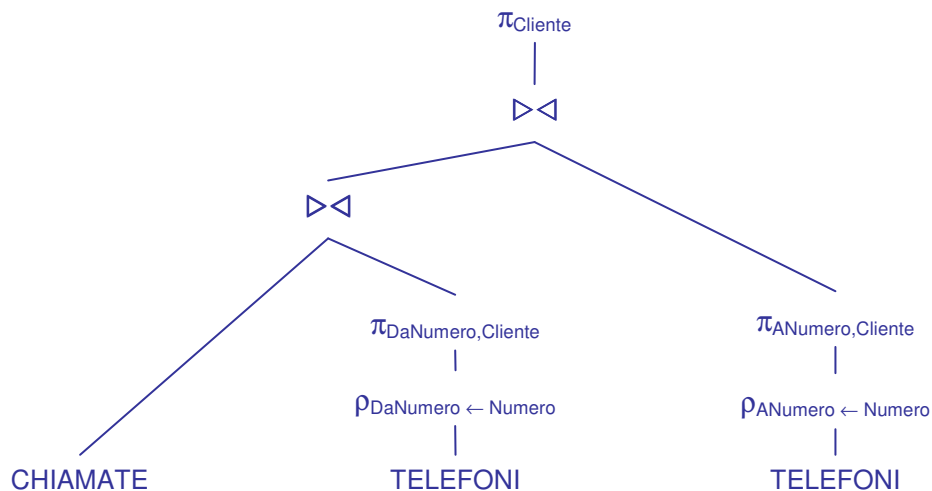
```
TELEFONI (Numero, Cliente, Saldo);  
CHIAMATE (DaNumero, ANumero, Inizio, Durata, Costo),  
DaNumero REFERENCES Telefoni,  
ANumero REFERENCES Telefoni;  
-- Inizio è di tipo TIMESTAMP (data, ora, minuti e secondi)  
-- Durata è un intero (secondi della conversazione)  
-- Saldo e Costo sono in formato DEC(6,2)
```

si scrivano in algebra relazionale le seguenti interrogazioni:

- 1.1) [1 p.]** Le chiamate effettuate il 21/02/2013 di durata inferiore ai 30 secondi eseguite da un telefono con saldo inferiore ai 30 €



- 1.2) [2 p.]** I clienti che hanno fatto almeno una chiamata tra due numeri intestati entrambi a loro



Sistemi Informativi T
21 febbraio 2013
Risoluzione

2) SQL (5 punti totali)

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

- 2.1) [2 p.]** I clienti che hanno eseguito almeno tre chiamate a numeri intestati a loro (i numeri chiamanti e chiamati non devono essere necessariamente gli stessi nelle tre o più chiamate)

```
SELECT TDA.CLIENTE
FROM TELEFONI TDA, TELEFONI TA, CHIAMATE C
WHERE C.DANUMERO = TDA.NUMERO
AND C.ANUMERO = TA.NUMERO
AND TDA.CLIENTE = TA.CLIENTE
GROUP BY TDA.CLIENTE
HAVING COUNT(*) >= 3
```

- 2.2) [3 p.]** Per ogni fascia di durata di conversazione (0-29: fascia 0, 30-59: fascia 30, 60-89: fascia 60, ecc.) si determini il numero di telefono che ha eseguito più chiamate, fornendo anche il relativo cliente, il saldo e la somma totale spesa nella fascia

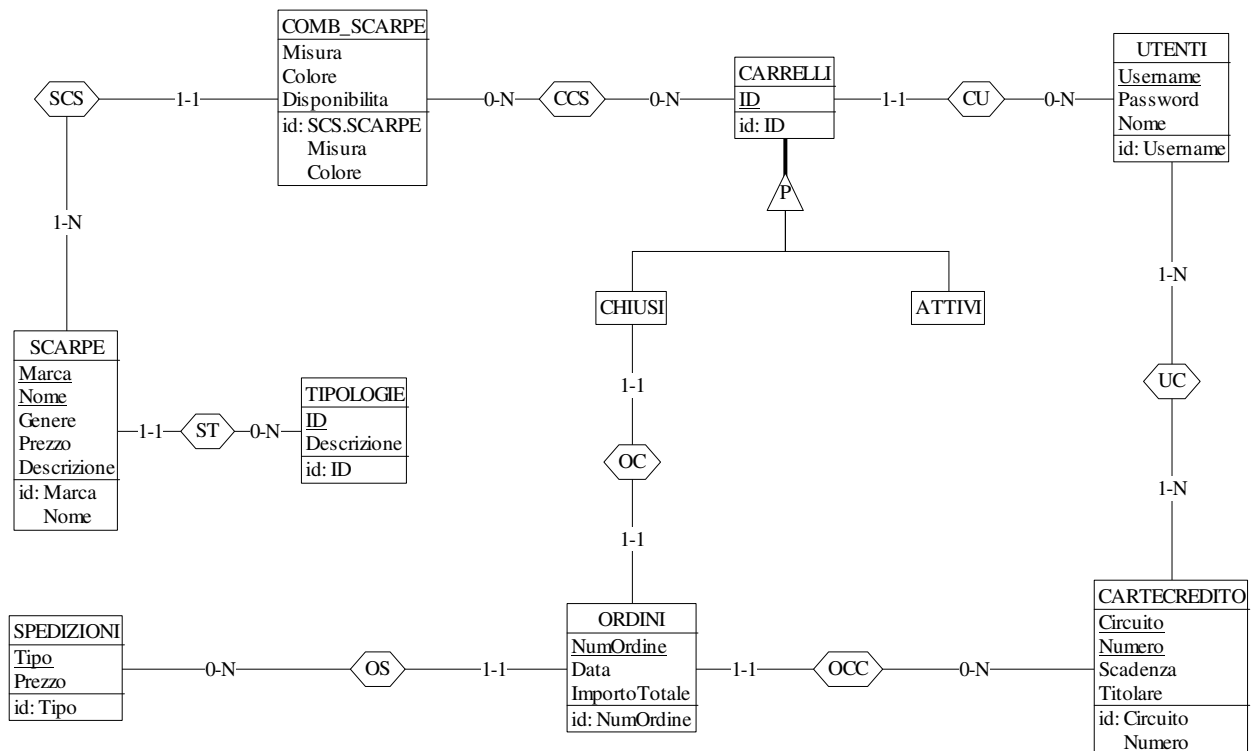
```
WITH FASCE(FASCIA,NUMERO,NUMCHIAMATE,TOTCOSTO) AS (
SELECT 30*(C.DURATA/30), C.DANUMERO, COUNT(*),SUM(COSTO)
FROM CHIAMATE C
GROUP BY C.DANUMERO, C.DURATA/30)
```

```
SELECT F1.FASCIA,F1.TOTCOSTO,T.*
FROM FASCE F1,TELEFONI T
WHERE F1.NUMERO = T.NUMERO
AND F1.NUMCHIAMATE = (SELECT MAX(F2.NUMCHIAMATE)
FROM FASCE F2
WHERE F2.FASCIA = F1.FASCIA)
ORDER BY F1.FASCIA
```

```
-- Nella common table expression è fondamentale raggruppare su
-- C.DURATA/30 per poter suddividere le chiamate in fasce di 30 secondi
```

3) Progettazione concettuale (6 punti)

Il negozio di scarpe on-line CamminaBene (CB) mette a disposizione modelli per tutti i gusti. Ogni scarpa, classificata per genere (uomo, donna, bambino e bambina) e tipologia (stivali, mocassini, sneakers, ecc.), ha una marca e un nome che la identificano, oltre a un prezzo e una descrizione, ed è presente in catalogo in diverse combinazioni di misura e colore. Per ogni combinazione è anche riportata la disponibilità in magazzino. CB permette a ogni utente registrato di scegliere prodotti e inserirli in uno o più "carrelli della spesa" virtuali, che vengono mantenuti attivi sino a che il cliente procede con l'ordine o li elimina esplicitamente. Nel primo caso il contenuto del carrello diventa quello dell'ordine (ogni ordine è relativo a un solo carrello e viceversa), nel secondo il sistema non ne tiene più traccia. Ogni pagamento può essere fatto solo con carta di credito, e CB tiene traccia dei dati (circuito, numero, scadenza e titolare) di tutte le carte usate dagli utenti, anche più di una per ogni utente. Nel caso di servizi di spedizione diversi da quello standard, il prezzo viene maggiorato di una quantità che dipende solo dal tipo di spedizione scelta, ma non dall'importo dell'ordine.

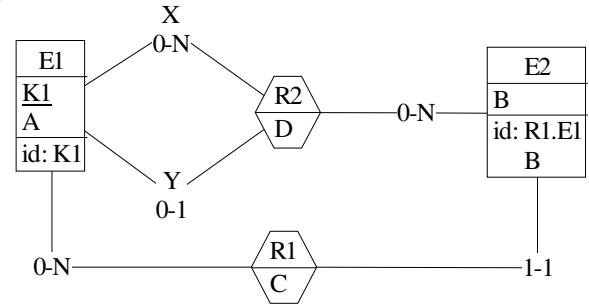
**Commenti:**

- A differenza del genere, le tipologie di scarpe sono in numero non specificato nel testo, e realisticamente variabile nel tempo, e quindi è opportuno definire un'entità.
- L'entità ATTIVI può essere omessa senza alterare la correttezza dello schema.
- E' opportuno modellare gli ORDINI con un'entità, permettendo così anche di esprimere l'univocità dei numeri d'ordine.
- In generale il titolare di una carta di credito può non coincidere con il cliente che la usa, così come la stessa carta può essere usata da utenti diversi.
- Lo schema ER non può esprimere il vincolo che la carta di credito usata per pagare un ordine sia del cliente che ha eseguito l'ordine stesso.
- Si notino le entità SCARPE e COMB_SCARPE, esempio del pattern LIBRO-COPIA_LIBRO visto a lezione (SCARPE è il concetto astratto, che si istanzia nell'entità COMB_SCARPE in qualcosa di concreto caratterizzato da Misura, Colore e Disponibilit ).

4) Progettazione logica (6 punti totali)

Dato lo schema concettuale in figura e considerando che:

- tutti gli attributi sono di tipo INT;
- nessuna associazione viene tradotta separatamente;
- ogni istanza di R2 associa istanze di E1 ed E2 con valori di K1 diversi tra loro;
- le istanze di E2 identificate esternamente dalla stessa istanza di E1 hanno valori di B la cui somma non può superare 100;



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 E2 (
  K1 INT NOT NULL,
  B INT NOT NULL,
  C INT NOT NULL,
  PRIMARY KEY (K1,B) );

CREATE TABLE E1 (
  K1 INT NOT NULL PRIMARY KEY,
  A INT NOT NULL,
  K1X INT REFERENCES E1,
  K1E2 INT,
  BE2 INT,
  D INT,
  CONSTRAINT R2_DEFINED CHECK
    ( (K1X IS NOT NULL AND K1E2 IS NOT NULL AND BE2 IS NOT NULL AND D IS NOT NULL) OR
      (K1X IS NULL AND K1E2 IS NULL AND BE2 IS NULL AND D IS NULL) ),
  CONSTRAINT FKR2 FOREIGN KEY (K1E2,BE2) REFERENCES E2,
  CONSTRAINT PUNTO_C CHECK ( K1 <> K1X AND K1 <> K1E2 AND K1X <> K1E2 ) );

ALTER TABLE E2
ADD CONSTRAINT FKR1 FOREIGN KEY (K1) REFERENCES E1;
```

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 ';')

-- Per garantire il rispetto del vincolo di cui al punto d) è necessario definire il seguente trigger:

```
CREATE TRIGGER PUNTO_D
BEFORE INSERT ON E2
REFERENCING NEW AS N
FOR EACH ROW
WHEN (100 < N.B + ( SELECT SUM(E2.B)
                     FROM E2
                     WHERE N.K1 = E2.K1 ) )
SIGNAL SQLSTATE '70001' ('La tupla inserita ha un valore di B troppo grande!');
```