Sistemi Informativi T
24 febbraio 2011

Tempo a disposizione: 2:30 ore

La consegna deve essere eseguita mediante l'apposito applicativo Web, facendo l'upload dei file specificati sul sito **http://esamix.labx** (solo per l'es. 1 la consegna è su carta)

N.B. Per superare la prova è necessario totalizzare almeno 3 punti negli esercizi 1 e 2

1) **Algebra relazionale (3 punti totali):**
Consegnare le risposte su un foglio di carta, intestato con matricola, nome e cognome

Date le seguenti relazioni, disponibili nello schema **b16884** con dati fittizi di esempio:

```
CLIENTI(CodC,Nome,Cognome,Comune);
ELETTRODOMESTICI(CodE,CodCliente,Marca,Modello,Tipologia),
CodCliente REFERENCES Clienti;
RIPARAZIONI(CodRip,CodE,Diagnosi,DataConsegna,DataRitiro*,Importo*),
CodE REFERENCES ELETTRODOMESTICI;
-- Importo e' un intero
-- DataRitiro e Importo hanno valore nullo per le riparazioni in corso
```

si scrivano in algebra relazionale le seguenti interrogazioni:

1.1) [1 p.] Tutte le riparazioni relative a clienti di Bologna e elettrodomestici di marca 'Revooh' con data di consegna posteriore al 1 gennaio 2011 e non ancora terminate

1.2) [2 p.] I codici degli elettrodomestici di tipo 'lavatrice' che hanno avuto la diagnosi 'perdita acqua' ma mai quella di 'guarnizione difettosa'

---

2) **SQL (5 punti totali)**
Consegnare il file SQL.txt

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

2.1) [2 p.] Per ogni cliente di Bologna, l'importo totale speso per riparazioni di durata superiore ai 7 giorni

2.2) [3 p.] Per ogni marca di lavatrici, l'importo più frequentemente pagato e le diagnosi ad esso associate

NB: Per misurare la differenza in giorni tra due date, queste vanno prima convertite mediante la funzione DAYS( ), ossia DAYS(date1) - DAYS(date2).
Il server in Lab3 accetta date nel formato 'DD/MM/YYYY'

Sistemi Informativi T
24 febbraio 2011

3) Progettazione concettuale (6 punti)

Consegnare il file ER.lun

Il sito ScacchiPerTutti permette di giocare gratuitamente a scacchi contro qualsiasi altro utente registrato sul sito, e identificato solamente dal suo nickname.
Di ogni partita si registra chi due giocatori gioca con i pezzi bianchi e chi con i neri, e il risultato finale (vittoria bianco, vittoria nero, o patta). Si tiene anche traccia di ogni mossa, identificata da un numero progressivo e da chi l'ha giocata (es. mossa bianco n. 1: Cf3, ossia cavallo in casella f3).
Ad ogni giocatore viene assegnato un punteggio iniziale di 1000 punti che viene variato a seconda dei risultati delle partite. Per ogni partita il sistema calcola (e inserisce nel DB) quanti punti ognuno dei due giocatori può vincere o perdere in funzione dell'esito della partita (intuitivamente, se un giocatore con punteggio basso gioca contro un giocatore con punteggio più alto, allora se vince guadagna più punti di quanti ne perderebbe in caso di sconfitta).
Il sistema permette a qualunque utente registrato di inserire commenti sulle partite o sulle singole mosse di una partita (es. commento dell'utente "nick1234" sulla quarta mossa del bianco nella partita n. 4157298: "Io avrei mangiato il cavallo!"). Ogni commento è caratterizzato da un identificatore univoco e da un timestamp.

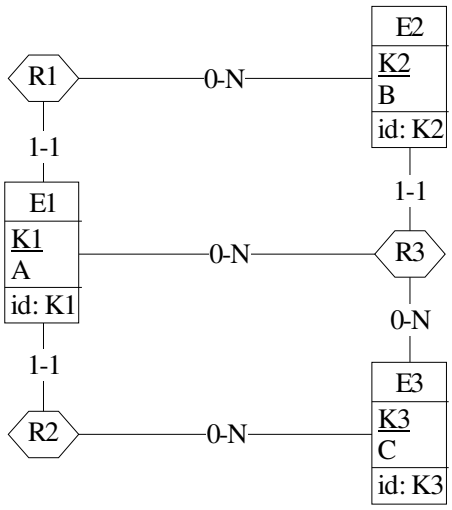
---

4) Progettazione logica (6 punti totali)

Consegnare i file SCHEMI.txt e TRIGGER.txt

Dato lo schema concettuale in figura
e considerando che:

a) tutti gli attributi sono di tipo INT;
b) le associazioni R1, R2 e R3 non vengono tradotte separatamente;
c) un'istanza di E1 non è mai associata, tramite R1 e R2, a una coppia di istanze (k2,k3) di E2 ed E3 che sono tra loro associate tramite R3;



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

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

IMPORTANTE:

- I file **NON** devono includere istruzioni di (dis)connessione al DB e contenere, alla fine del file TRIGGER.txt, il **DROP** degli oggetti creati
- Per il punto 4.2), se necessario, si specifichino usando commenti SQL eventuali inserimenti di tipo transazionale (ossia, più INSERT nella stessa transazione)
- La risoluzione del punto 4.2) può avvenire anche specificando semplicemente equivalenti "query di verifica" da eseguire prima degli inserimenti; in tal caso si ha 1 solo punto a disposizione
- Si prega di attenersi scrupolosamente alle istruzioni relative ai nomi dei file (maiuscole incluse), in quanto gli script verranno testati automaticamente. **Il mancato rispetto delle istruzioni comporterà penalizzazioni di punteggio**