Sistemi Informativi T
6 febbraio 2017

Tempo a disposizione: 2:30 ore
(Esame di SI L-A: esercizi 1) e 2), 1 ora; Esame di SI L-B: esercizi 3) e 4), 1:30 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 di SI-T 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:

```
PRODOTTI(CodP,Descrizione,Prezzo,NumVenduti);
PACCHETTI(PID,Prezzo,Venduti);
COMPOSIZIONI(PID,CodP,NumUnita),
      PID REFERENCES PACCHETTI,
      CodP REFERENCES PRODOTTI;
--
-- Ogni pacchetto (o "bundle") consiste di uno o più prodotti,
-- ciascuno in quantità specificata da NumUnita.
-- Il Prezzo di un pacchetto è sempre inferiore a quello che si
-- pagherebbe comprando i vari prodotti separatamente.
-- NumVenduti e Venduti indicano quante unità di prodotto o
-- pacchetto, rispettivamente, sono state complessivamente vendute.
-- NB: NumVenduti tiene conto anche delle unità vendute all'interno
-- di pacchetti di cui il prodotto fa parte, oltre che delle unità
-- relative alle vendite del singolo prodotto.
-- Tutti i prezzi sono in formato DEC(6,2).
```

si scrivano in algebra relazionale le seguenti interrogazioni:

1.1) [1 p.] I dati dei pacchetti venduti almeno 10 volte in cui c'è almeno una 'TV' che costa meno di 500€ e complessivamente venduta 40 o più volte

1.2) [2 p.] I codici dei pacchetti che hanno almeno due prodotti in cui il prezzo singolo di uno è meno della metà dell'altro

---

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 codice di prodotto, il numero di unità vendute singolarmente (ossia levando quelle vendute nei pacchetti). Il risultato va ordinato per valori decrescenti di questa quantità, escludendo i prodotti presenti in un solo pacchetto

2.2) [3 p.] Il codice del prodotto per cui è minimo l'incasso totale di tutti pacchetti in cui il prodotto non è presente

Sistemi Informativi T
6 febbraio 2017

3) Progettazione concettuale (6 punti)

Consegnare il file ER.lun

Il social network MeetUs è pensato per permettere ai suoi utenti di organizzare eventi di varia natura e condividerli con gli altri utenti del network. Ogni utente, identificato da un nickname, ha una password e un indirizzo di posta elettronica (lo stesso indirizzo non può essere usato per creare più utenti). Quando un utente crea un evento deve specificarne la tipologia (culturale, musicale, festa, ecc.), scegliendo tra le opzioni previste da MeetUs, il luogo, la data e l'orario di inizio. Può inoltre specificare se l'evento è riservato ai soli iscritti a certi gruppi. Un gruppo di MeetUs è creato da un utente, che ne diventa automaticamente membro. Per ogni iscritto a un gruppo si tiene traccia della relativa data di iscrizione.
Per ogni evento pubblicato, un utente può specificare se parteciperà o meno, e MeetUs registra il timestamp dell'eventuale scelta. Di un evento passato si archiviano solo le caratteristiche e il numero di utenti che hanno scelto di partecipare.

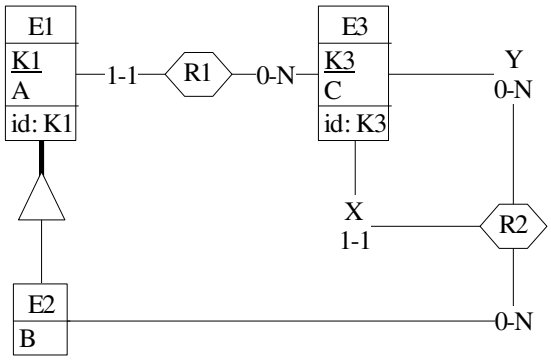
---

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 entità E1 ed E2 vengono tradotte assieme
c) le associazioni R1 e R2 non vengono tradotte separatamente;
d) un'istanza di E1 non può essere associata, tramite R1, a un'istanza di E3 che partecipa all'associazione R2 con ruolo Y;



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 singole tuple non corrette**, definiti in un file TRIGGER.txt e usando se necessario il simbolo '@' per terminare gli statement SQL (altrimenti ';')

IMPORTANTE:

- I file **NON** devono includere istruzioni di (dis)connessione al DB
- Per il punto 4.2), se necessario, si specifichino usando commenti SQL eventuali inserimenti di tipo transazionale (ossia, più INSERT nella stessa transazione)
- Si prega di attenersi scrupolosamente alle istruzioni relative ai nomi dei file (maiuscole incluse). **Il mancato rispetto delle istruzioni potrà comportare penalizzazioni di punteggio**