Sistemi Informativi T
3 luglio 2020

Tempo a disposizione: 3:00 ore

Consegnare i file specificati per gli esercizi 2, 3 e 4.
Per l'es. 1 fare una foto del foglio con le soluzioni e quindi caricare il file ottenuto.

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):**
Fotografare il foglio di carta con le risposte e consegnare il file ALGEBRA.<fmt>
(<fmt> è l'estensione del formato scelto, ad es. jpg)

Date le seguenti relazioni, definite nello schema **B16884** ma prive di dati:

```
VESTITI(VCod,Marca,Descrizione,Prezzo);
DISPONIBILITA(TCID,VCod,Taglia,Colore,NumDisponibili),
      VCod REFERENCES VESTITI;
VENDITE(TCID,Data,Numero),
      TCID REFERENCES DISPONIBILITA;
--
-- Prezzo è di tipo DEC(6,2).
-- Taglia ha valori stringa ('XXS','XS','S','M','L',….).
-- NumDisponibili è un intero non negativo (0 = non disponibile)
-- Numero è un intero > 0 che indica il numero di vestiti
-- di un dato colore e una data taglia venduti in un dato giorno.
```

si scrivano in algebra relazionale le seguenti interrogazioni:

1.1) [1 p.] Le date in cui si sono incassati più di 2000€ per almeno un vestito di marca 'IVEL' di colore bianco e taglia 'XL'

1.2) [2 p.] I (codici di) vestiti della marca 'EKIN' che nel 2020 non hanno registrato nessuna vendita

---

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

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

2.1) [2 p.] Per ogni vestito, la quantità complessivamente disponibile in magazzino, escludendo le combinazioni (Taglia,Colore) che nel 2020 non sono mai state vendute

2.2) [3 p.] Per ogni vestito, la combinazione (Taglia,Colore) che ha incassato di più nel 2019

NB: Per l'uso delle funzioni SQL relative a date, orari e altro si consulti il file FunzioniSQL nella pagina del Lab

3) Progettazione concettuale (6 punti)
Consegnare il file ER.lun

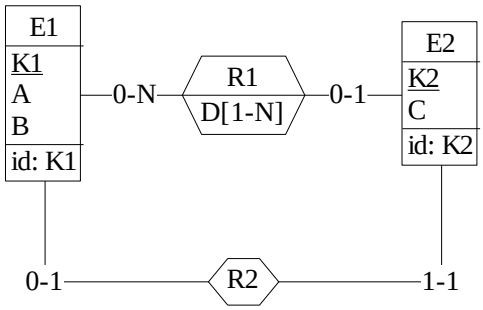
Le Olimpiadi di Programmazione (OP) permettono l'iscrizione di squadre di studenti universitari da tutto il mondo. Ogni squadra, identificata dal nome della propria Università, si compone di un numero di studenti variabile, da 6 fino a un massimo di 10, per ognuno dei quali si riportano i principali dati anagrafici (nome, cognome, luogo e data di nascita, ed email identificativa). Le OP prevedono una serie di prove, ognuna caratterizzata da un tempo a disposizione e dai punteggi assegnati alle prime 3 squadre (ad es. la prova 5: "algoritmo di chiusura transitiva su un grafo orientato" ha una durata di 180 minuti, 10 punti per la prima squadra, 5 per la seconda e 2 per la terza). Per ogni prova viene anche dichiarato quanti studenti di ogni squadra possono svolgere quella prova, e quando la prova viene effettivamente svolta si registrano gli studenti che l'hanno svolta e si memorizza il codice consegnato.
Il DB delle OP fornisce inizialmente tutti i dettagli che caratterizzano le varie prove (nome, durata, punteggi, ecc.), quindi i dettagli delle squadre partecipanti, e man mano gli esiti delle diverse prove (chi l'ha svolta, la posizione ottenuta e il codice consegnato).

---

4) Progettazione logica (6 punti totali)
Consegnare i file SCHEMI.txt e TRIGGER.txt (o SCHEMI.sql e TRIGGER.sql)

Dato lo schema concettuale in figura e considerando che:

a) tutti gli attributi sono di tipo INT;
b) nessuna associazione viene tradotta separatamente (in particolare, R2 viene tradotta con E2);
c) vale la dipendenza funzionale C → B, ovvero a valori uguali di C in due istanze di E2 corrispondono, tramite R2, due istanze di E1 con lo stesso valore di B;



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 (o SCHEMI.sql)

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 (o TRIGGER.sql) 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**