

Sistemi Informativi T
17 gennaio 2012

Tempo a disposizione: 2:30 ore

SI L-A: (solo esercizi 1) e 2)) 1 ora;

SI L-B: (solo 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:

```
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
```

si scrivano in algebra relazionale le seguenti interrogazioni:

- 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
-

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.]** Codice, nome e cognome dei soci che non hanno usato nessuna barca negli ultimi 100 giorni
- 2.2) [3 p.]** La lunghezza di barca che è stata prenotata complessivamente il maggior numero di volte

NB: L'espressione YEAR(Data) restituisce un anno;
per calcolare i giorni trascorsi tra due date (Data1 e Data2), fare uso dell'espressione DAYS(Data1) - DAYS(Data2);
il registro di DB2 CURRENT DATE fornisce la data attuale

3) Progettazione concettuale (6 punti)

*Consegnare il file **ER.lun***

Il sistema di prenotazioni mediche on-line SanaWeb permette a tutti gli **utenti** del Servizio Sanitario Nazionale (SSN), identificati da un codice SSN e descritti dai loro dati anagrafici (nome, cognome, indirizzo, luogo e data di nascita, sesso) di prenotare visite specialistiche presso le **strutture** (pubbliche o private) che aderiscono al servizio. Ogni struttura, di cui sono noti nome, indirizzo e recapiti telefonici (uno o più), mette a disposizione una serie di **servizi** tra quelli complessivamente prenotabili via SanaWeb (ogni servizio è offerto da almeno una struttura). Per le strutture private va mantenuta anche la data della convenzione con il SSN.

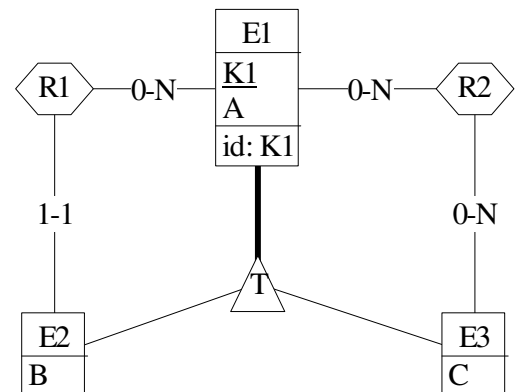
Ogni **prenotazione** ha una data, un orario e dettagli sul luogo specifico all'interno della struttura (es. laboratorio analisi IV piano stanza 17). Per le prenotazioni il cui relativo servizio è stato **erogato** (ci sono anche prenotazioni che vengono **cancellate** e altre relative a date **future**), SanaWeb mette a disposizione la possibilità di consultare on-line i relativi **documenti prodotti**. Per ogni documento va specificato il tipo (ad es. esiti esami, referti, radiografie; i possibili **tipi sono predefiniti** da SanaWeb), e fornito il **relativo file, il cui nome identifica univocamente il documento**.

4) Progettazione logica (6 punti totali)

*Consegnare i file **SCHEMI.txt** e **TRIGGER.txt***

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**

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**