

Sistemi Informativi T
19 febbraio 2020

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 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:

```
FURGONI (Targa, Sede) ;  
VIAGGI (IDV, Targa, Data, KmTotali) ,  
      Targa REFERENCES FURGONI ;  
TAPPE (IDV, Num, Km) ,  
      IDV REFERENCES VIAGGI ;
```

```
-- KmTotali, Km e Num sono di tipo INT.  
-- L'attributo TAPPE.Num è un progressivo che identifica le tappe di  
-- uno stesso viaggio.  
-- L'attributo TAPPE.Km, compreso tra 0 e KmTotali, indica a quanti  
-- chilometri dalla partenza è stata fatta una tappa.  
-- Per semplicità, oltre alle tappe "vere e proprie" (o intermedie),  
-- TAPPE include per ogni viaggio anche 2 tuple, per la partenza  
-- (tappa num. 0, Km 0) e per l'arrivo (Km = KmTotali), che sono quindi  
-- le sole per un viaggio senza tappe intermedie.
```

si scrivano in algebra relazionale le seguenti interrogazioni:

- 1.1) [1 p.]** I codici dei viaggi dei furgoni della sede di Modena in cui c'è stata almeno una tappa intermedia dopo la metà del viaggio, e il viaggio era di almeno 700 km
- 1.2) [2 p.]** Le targhe dei furgoni che non hanno mai fatto una tappa intermedia in viaggi di meno di 400 km

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 fascia di chilometri totali (0-399,400-799, ecc.) si determini la targa del furgone che ha effettuato più viaggi in quella fascia, e si ordini il risultato per fascia
- 2.2) [3 p.]** Considerando solo i viaggi con almeno una tappa intermedia, si determini per ogni viaggio la lunghezza massima di una tappa
(ad es., se un viaggio ha tappe a 0, 350, 600 e 1000 km, la tappa più lunga è di $1000-600 = 400$ km)

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

L'istituto di vigilanza Tranquillo 24h (T24H) assicura il controllo a distanza di qualsiasi impianto d'allarme installato. T24H mantiene per ogni impianto d'allarme i dati del modello (ogni modello ha una marca, un nome e un codice univoco per quella marca), la data d'installazione e altri dettagli. Viene ovviamente registrato il tipo di edificio (abitazione in condominio, villa, capannone, ecc., scelto da un elenco predefinito), l'indirizzo, il nome del cliente e uno o più recapiti telefonici (un cliente può essere titolare di più sistemi di allarme). In caso di allarme ricevuto, l'operatore della T24H registra data e ora e contatta il cliente per ulteriori informazioni. Sulla base di quanto appurato, T24H può avviare un intervento inviando una pattuglia, oppure inoltrare la richiesta ai Vigili del Fuoco e/o alle Forze dell'Ordine. Nel caso di intervento si registrano la targa dell'auto di pattuglia inviata e i dipendenti a bordo dell'auto (ogni dipendente ha un codice, nome, cognome e un numero di telefono).

4) Progettazione logica (6 punti totali)

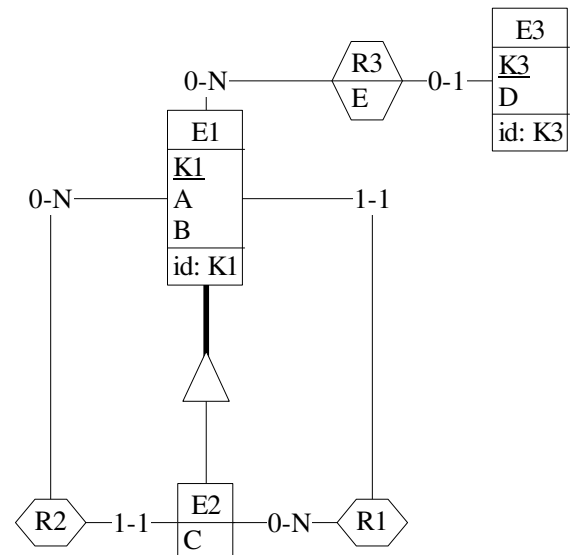
Consegnare i file SCHEMI.txt e TRIGGER.txt (o SCHEMI.sql e TRIGGER.sql)

Dato lo schema concettuale in figura e considerando che:

- tutti gli attributi sono di tipo INT;
- le entità E1 ed E2 vengono tradotte insieme;
- nessuna associazione viene tradotta separatamente;
- un'istanza di E3 non è mai associata a un'istanza di E1 che partecipa all'associazione R2 dal ramo 0-N;

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**