

**Sistemi Informativi T**  
**14 gennaio 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 (Targa, Data, OraPartenza, OraArrivo, KmPercorsi, LitriUsati) ,
      Targa REFERENCES FURGONI ;
RIFORNIMENTI (Targa, Data, Ora, NumLitri) ,
      Targa REFERENCES FURGONI ;

-- OraPartenza, OraArrivo e Ora sono di tipo TIME (es. '12:32').
-- NumLitri e LitriUsati sono di tipo DEC(6,2).
-- KmPercorsi è di tipo INT.
-- Per semplicità si suppone che ogni viaggio inizi e termini in uno
-- stesso giorno.
```

si scrivano in algebra relazionale le seguenti interrogazioni:

- 1.1) [1 p.]** Le targhe dei furgoni della sede di Bologna che nel 2019 hanno fatto un viaggio di almeno 500 km facendo nello stesso giorno un rifornimento di meno di 30 litri
- 1.2) [2 p.]** Le targhe dei furgoni che, in ogni viaggio fatto nel 2019, hanno sempre fatto almeno un rifornimento nello stesso giorno del viaggio

---

**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 sede, il consumo medio complessivo (espresso in km/litro) dei propri furgoni nel 2020, escludendo i viaggi in cui è stato fatto rifornimento durante il viaggio stesso (ovvero, dopo la partenza e prima dell'arrivo)  
NB: consumo medio: si calcoli il consumo in km/litro per ogni viaggio e si faccia la media
- 2.2) [3 p.]** Considerando in ogni anno per ogni furgone solo il relativo viaggio di maggior durata (per ipotesi unico), si determini per ogni anno la targa del furgone che ha avuto il minimo consumo in km/litro

**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*

La redazione del settimanale IeriOggiDomani (IOD) si occupa della gestione delle inserzioni pubblicitarie sui numeri della rivista.

Per ogni inserzionista, che può essere un privato cittadino o un ente, si memorizza l'indirizzo, il numero di telefono e le aree d'interesse (ogni area è caratterizzata da un codice d'area e da un nome). Per ogni privato si mantengono inoltre nome, cognome e codice fiscale, mentre per gli enti ragione sociale e partita IVA.

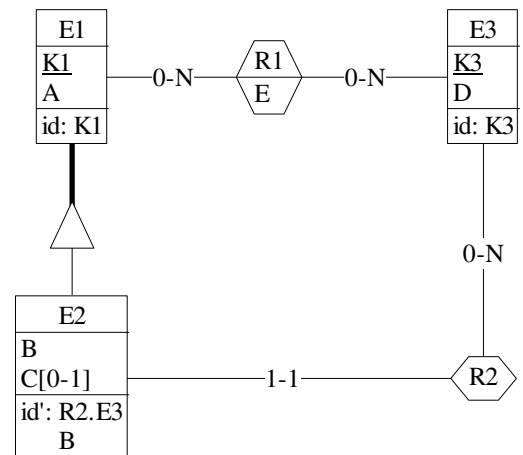
Per ogni inserzione, caratterizzata dalla dimensione (larghezza, altezza), da un titolo descrittivo e dal riferimento a un'immagine, IOD stabilisce un prezzo in funzione del numero di volte che viene pubblicata sulla rivista. Gli enti possono anche concordare con IOD delle "campagne pubblicitarie", ovvero più inserzioni legate da un tema comune (ad es. prodotti di una stessa linea). Nel caso di inserzioni facenti parte di una campagna il prezzo non è dato (in quanto viene riferito alla campagna nel suo complesso) e le inserzioni vengono identificate anche dalla campagna di cui fanno parte e da un numero univoco all'interno della campagna stessa.

**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) le entità E1 ed E2 vengono tradotte insieme;
- c) un'istanza di E2 non è mai associata, tramite R1, all'istanza di E3 che la identifica esternamente;



**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**