Sistemi Informativi T
14 gennaio 2020
Risoluzione

Tempo a disposizione: 2:30 ore

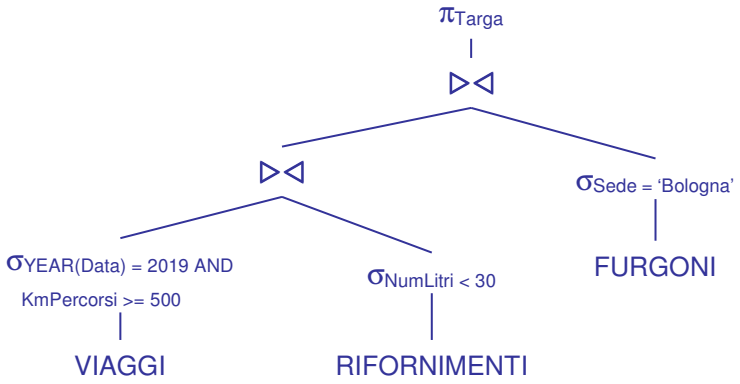
1) Algebra relazionale (3 punti totali):

Date le seguenti relazioni:

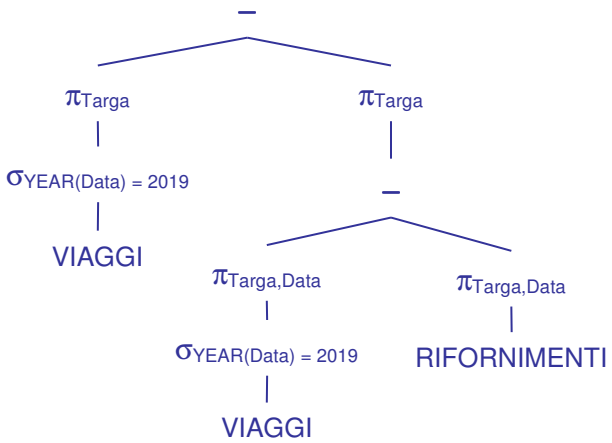
```
FURGONI(Targa,Sede);
VIAGGI(Targa,Data,OraPartenza,OraArrivo,KmPercorsi,LitriUsati),
      Targa REFERENCES FURGONI;
RIFORNIMENTI(Targa,Data,Ora,NumLitri),
      Targa REFERENCES FURGONI;

-- 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



La prima differenza trova i furgoni che in almeno un viaggio del 2019 non hanno fatto rifornimento lo stesso giorno

Sistemi Informativi T
14 gennaio 2020
Risoluzione

SQL (5 punti totali)

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

```
SELECT   F.Sede, DEC(AVG(KmPercorsi/LitriUsati),6,2) AS CONSUMO_MEDIO
FROM     FURGONI F, VIAGGI V
WHERE    F.Targa = V.Targa
AND      YEAR(V.Data) =2020
AND      NOT EXISTS (SELECT *
                     FROM RIFORNIMENTI R
                     WHERE (R.Targa,R.Data) = (V.Targa,V.Data)
                     AND R.Ora BETWEEN V.OraPartenza AND V.OraArrivo )
GROUP BY F.Sede;
```

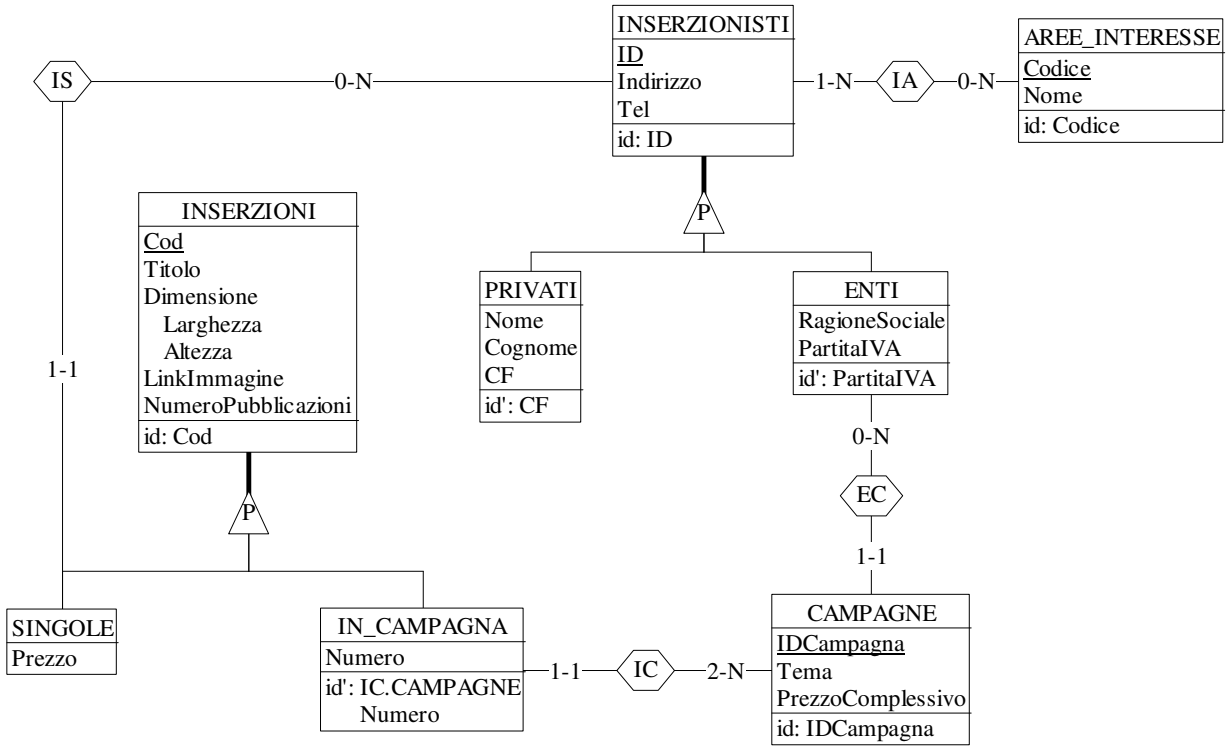
2.2) **[3 p.]** Considerando in ogni anno per ogni furgone solo il relativo viaggio di maggior durata (per ipotesi unico), si determini per ogni anno la targa del furgone che ha avuto il minimo consumo in km/litro

```
WITH
VIAGGI_MAX_DURATA AS (
   SELECT *
   FROM   VIAGGI V
   WHERE  (V.OraArrivo-V.OraPartenza) =
          (     SELECT MAX(V1.OraArrivo-V1.OraPartenza)
                FROM   VIAGGI V1
                WHERE  V1.Targa = V.Targa
                AND    YEAR(V1.Data) = YEAR(V.Data) )
   )
SELECT   YEAR(V.Data) AS ANNO, V.Targa
FROM     VIAGGI_MAX_DURATA V
WHERE    V.KmPercorsi/LitriUsati =
   (     SELECT MAX(V1.KmPercorsi/LitriUsati)
         FROM   VIAGGI_MAX_DURATA V1
         WHERE  YEAR(V1.Data) = YEAR(V.Data) );
-- La c.t.e. seleziona per ogni anno e furgone il viaggio di durata
-- massima
```

Sistemi Informativi T
14 gennaio 2020
Risoluzione

3) Progettazione concettuale (6 punti)

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.



Commenti:

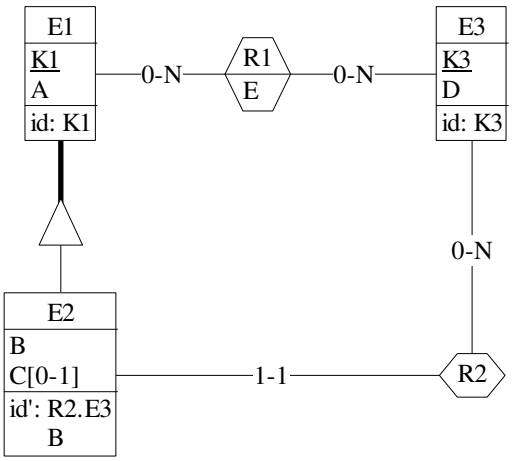
- La sola difficoltà nell'eserczio è data dal corretto collegamento tra le entità coinvolte nelle gerarchie.

Sistemi Informativi T
14 gennaio 2020
Risoluzione

4) Progettazione logica (6 punti totali)

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)

```
CREATE TABLE E3 (
K3      INT NOT NULL PRIMARY KEY,
D       INT NOT NULL                    );

CREATE TABLE E1 (
K1      INT NOT NULL PRIMARY KEY,
A       INT NOT NULL,
TIPO    SMALLINT NOT NULL CHECK (TIPO IN (1,2)),        -- 2: istanza anche di E2
B       INT,
C       INT,
K3      INT REFERENCES E3,
CONSTRAINT E2 CHECK ((TIPO = 1 AND B IS NULL AND C IS NULL AND K3 IS NULL) OR
                     (TIPO = 2 AND B IS NOT NULL AND K3 IS NOT NULL) )          );

CREATE TABLE R1 (
K1      INT NOT NULL REFERENCES E1,
K3      INT NOT NULL REFERENCES E3,
E       INT NOT NULL,
PRIMARY KEY (K1,K3)                     );
```

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 ';')

```
-- Trigger che garantisce che l'unicità delle coppie di valori (K3,B)
CREATE TRIGGER E2_KEY
BEFORE INSERT ON E1
REFERENCING NEW AS N
FOR EACH ROW
WHEN ( EXISTS (     SELECT *
                    FROM E1
                    WHERE (N.K3,N.B) = (K3,B) )    )
SIGNAL SQLSTATE '70001' ('La coppia (K3,B) non può essere duplicata!');

-- Trigger che garantisce il rispetto del vincolo al punto c)
CREATE TRIGGER PUNTO_C
BEFORE INSERT ON R1
REFERENCING NEW AS N
FOR EACH ROW
WHEN ( N.K3 =  (    SELECT K3
                    FROM E1
                    WHERE K1 = N.K1 )    )
SIGNAL SQLSTATE '70002' ('L''istanza di E2 è associata all''istanza di E3 che la identifica esternamente!');
```