

Sistemi Informativi T
19 febbraio 2020
Risoluzione

Tempo a disposizione: 2:30 ore

1) Algebra relazionale (3 punti totali):

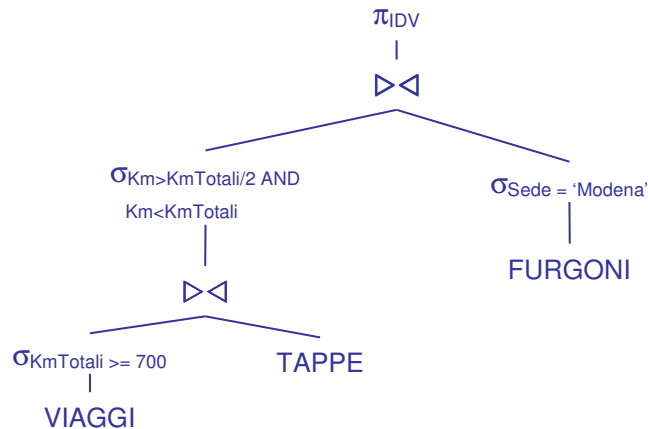
Date le seguenti relazioni:

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

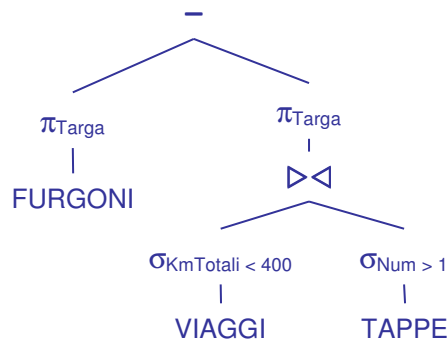
-- KmTotali, Km e Num sono di tipo INT.
-- L'attributo TAPPE.Num è un progressivo che identifica le tappe di
-- uno stesso viaggio.
-- L'attributo TAPPE.Km, compreso tra 0 e KmTotali, indica a quanti
-- chilometri dalla partenza è stata fatta una tappa.
-- Per semplicità, oltre alle tappe "vere e proprie" (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



Sistemi Informativi T
19 febbraio 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 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

```
WITH VIAGGI_PER_FASCIA (Targa,Fascia,NumViaggi) AS (  
    SELECT    V.Targa, V.KmTotali/400, COUNT(*)  
    FROM      VIAGGI V  
    GROUP BY V.Targa, V.KmTotali/400 )  
SELECT    VF.Fascia*400 || '--' || ((VF.Fascia+1)*400 - 1) AS Fascia, VF.Targa  
FROM      VIAGGI_PER_FASCIA VF  
WHERE     VF.NumViaggi =      ( SELECT MAX(VF1.NumViaggi)  
                                FROM    VIAGGI_PER_FASCIA VF1  
                                WHERE   VF1.Fascia = VF.Fascia )  
  
ORDER BY Fascia;  
  
-- La c.t.e. calcola per ogni targa e fascia il numero di viaggi di quel  
-- furgone in quella fascia
```

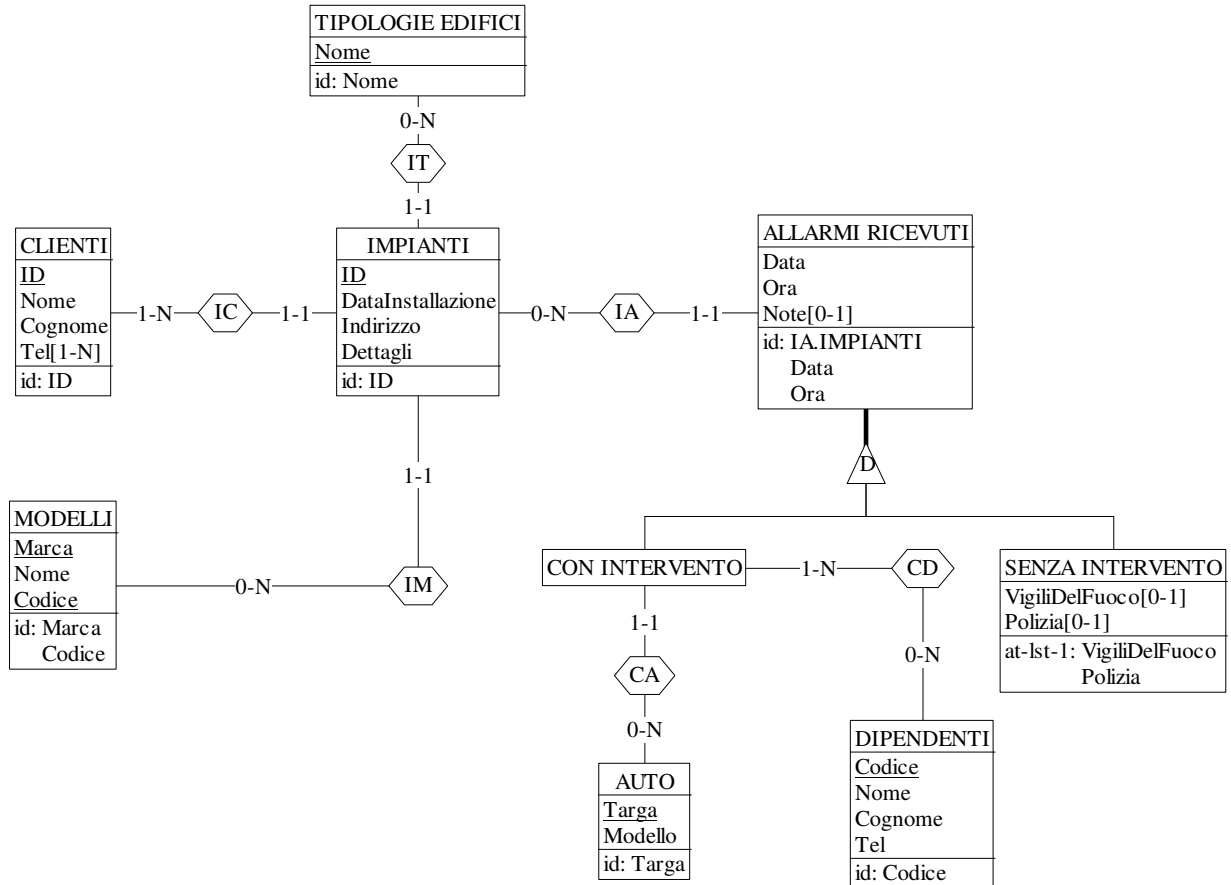
- 2.2) [3 p.]** Considerando solo i viaggi con almeno una tappa intermedia, si determini per ogni viaggio la lunghezza massima di una tappa

```
WITH LUNGHEZZA_TAPPE (IDV,Lung) AS (  
    SELECT TP.IDV, TA.Km - TP.Km  
    FROM    TAPPE TP, TAPPE TA  
    WHERE   TP.IDV = TA.IDV  
    AND     TP.Num = TA.Num - 1      -- tappe consecutive  
    AND     EXISTS ( SELECT *        -- viaggio con sosta intermedia  
                     FROM    TAPPE T  
                     WHERE   T.IDV = TP.IDV  
                     AND     T.Num > 1 ) )  
  
SELECT LT.*  
FROM    LUNGHEZZA_TAPPE LT  
WHERE   LT.Lung >= ALL ( SELECT LT1.Lung  
                        FROM    LUNGHEZZA_TAPPE LT1  
                        WHERE   LT1.IDV = LT.IDV ) );  
  
-- La c.t.e. determina la lunghezza delle tappe di ciascun viaggio che ha  
-- almeno una tappa intermedia (Num > 1)
```

Sistemi Informativi T
19 febbraio 2020
Risoluzione

3) Progettazione concettuale (6 punti)

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).



Commenti:

- Gli attributi di **SENZA INTERVENTO** non possono essere entrambi nulli (vincolo at-1st-1).
- La gerarchia è parziale perché la decisione di cosa fare avviene successivamente alla registrazione dell'allarme ricevuto.

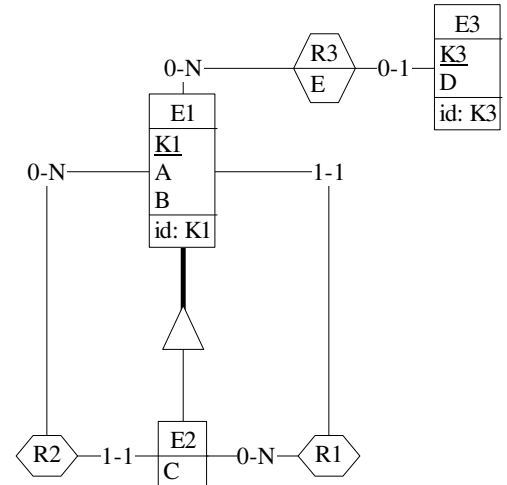
Sistemi Informativi T
19 febbraio 2020
Risoluzione

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) nessuna associazione viene tradotta separatamente;
- d) 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**)



```
CREATE TABLE E1 (
K1      INT NOT NULL PRIMARY KEY,
A       INT NOT NULL,
B       INT NOT NULL,
K1R1    INT NOT NULL REFERENCES E1,
TIPO    SMALLINT NOT NULL CHECK (TIPO IN (1,2)),
C       INT,
K1R2    INT REFERENCES E1,
CONSTRAINT E2 CHECK ((TIPO = 1 AND C IS NULL AND K1R2 IS NULL) OR
(TIPO = 2 AND C IS NOT NULL AND K1R2 IS NOT NULL))
);
```

-- 2: istanza anche di E2

```
CREATE TABLE E3 (
K3      INT NOT NULL PRIMARY KEY,
D       INT NOT NULL,
K1R3    INT REFERENCES E1,
E       INT,
CONSTRAINT R3 CHECK ((E IS NULL AND K1R3 IS NULL) OR
(E IS NOT NULL AND K1R3 IS NOT NULL))
);
```

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 ';')

```
CREATE TRIGGER R1
BEFORE INSERT ON E1
REFERENCING NEW AS N
FOR EACH ROW
WHEN ( NOT EXISTS ( SELECT *
FROM E1
WHERE N.K1R1 = E1.K1
AND E1.TIPO = 2 ))
SIGNAL SQLSTATE '70001' ('L'istanza di E1 deve referenziare un'istanza di E2!');
```

-- Il vincolo può essere violato inserendo in E1 o in E3; i trigger confrontano comunque i valori di 2 foreign key

```
CREATE TRIGGER PUNTO_D_E1
BEFORE INSERT ON E1
REFERENCING NEW AS N
FOR EACH ROW
WHEN ( EXISTS ( SELECT * FROM E3 WHERE N.K1R2 = E3.K1R3 ) )
SIGNAL SQLSTATE '70002' ('L'istanza di E1 referenziata da N.K1R2 partecipa all'associazione R3!);
```

```
CREATE TRIGGER PUNTO_D_E3
BEFORE INSERT ON E3
REFERENCING NEW AS N
FOR EACH ROW
WHEN ( EXISTS ( SELECT * FROM E1 WHERE N.K1R3 = E1.K1R2 ) )
SIGNAL SQLSTATE '70003' ('L'istanza di E1 referenziata da N.K1R2 partecipa all'associazione R2 lato 0-N!);
```