

Sistemi Informativi T
17 giugno 2014
Risoluzione

Tempo a disposizione: 2:30 ore

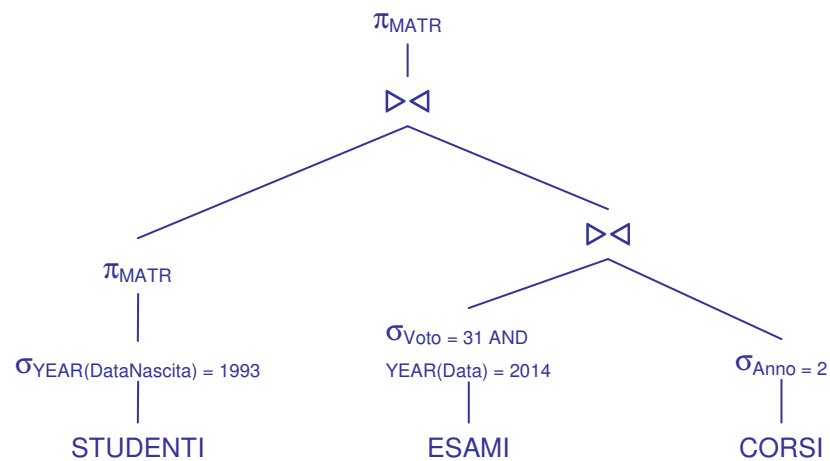
1) Algebra relazionale (3 punti totali):

Date le seguenti relazioni:

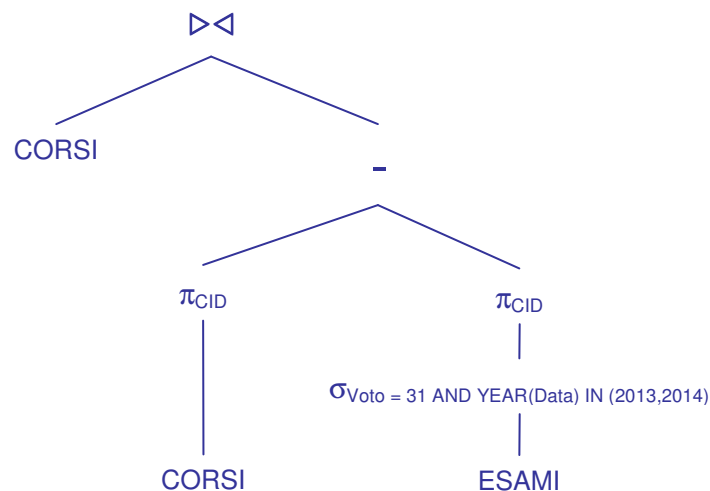
```
STUDENTI (MATR, Nome, Cognome, DataNascita);  
CORSI (CID, Nome, Anno, CFU);  
ESAMI (MATR, CID, Data, Voto),  
    MATR REFERENCES STUDENTI,  
    CID REFERENCES CORSI;  
--  
-- Voto ha valori compresi tra 18 e 31 (31 = 30 e lode)  
-- CFU e' un intero positivo
```

si scrivano in algebra relazionale le seguenti interrogazioni:

1.1) [1 p.] I numeri di matricola degli studenti nati nel 1993 che hanno preso almeno un 30 e lode nel 2014 in un corso del secondo anno



1.2) [2 p.] I dettagli dei corsi in cui non è mai stato dato un 30 e lode né nel 2013 né nel 2014



Sistemi Informativi T
17 giugno 2014
Risoluzione

2) SQL (5 punti totali)

Con riferimento al DB dell'esercizio 1, si scrivano in SQL le seguenti interrogazioni:

- 2.1) [2 p.]** Per ogni studente nato nel 1993 e per ogni anno di corso (CORSI.Anno), il numero totale di esami sostenuti

```
SELECT    S.MATR, C. Anno, COUNT(*) AS NumEsami
FROM      STUDENTI S, CORSI C, ESAMI E
WHERE     S.MATR = E.MATR
AND       E.CID = C.CID
AND       YEAR(DataNascita) = 1993
GROUP BY  S.MATR, C. Anno
```

- 2.2) [3 p.]** Per ogni fascia di CFU totali acquisiti (fascia 0: 0-49 CFU, fascia 1: 50-99 CFU, ecc.) il relativo numero di studenti nella fascia (es.: fascia 0: 29 studenti, fascia 1: 53 studenti, ecc.)

```
WITH TOTCFU (MATR, NumCFU) AS (
    SELECT  E.MATR, SUM(C.CFU)
    FROM    CORSI C, ESAMI E
    WHERE   E.CID = C.CID
    GROUP BY E.MATR
)

SELECT    NumCFU/50 AS Fascia, COUNT(*) AS NumStudenti
FROM      TOTCFU
GROUP BY  NumCFU/50

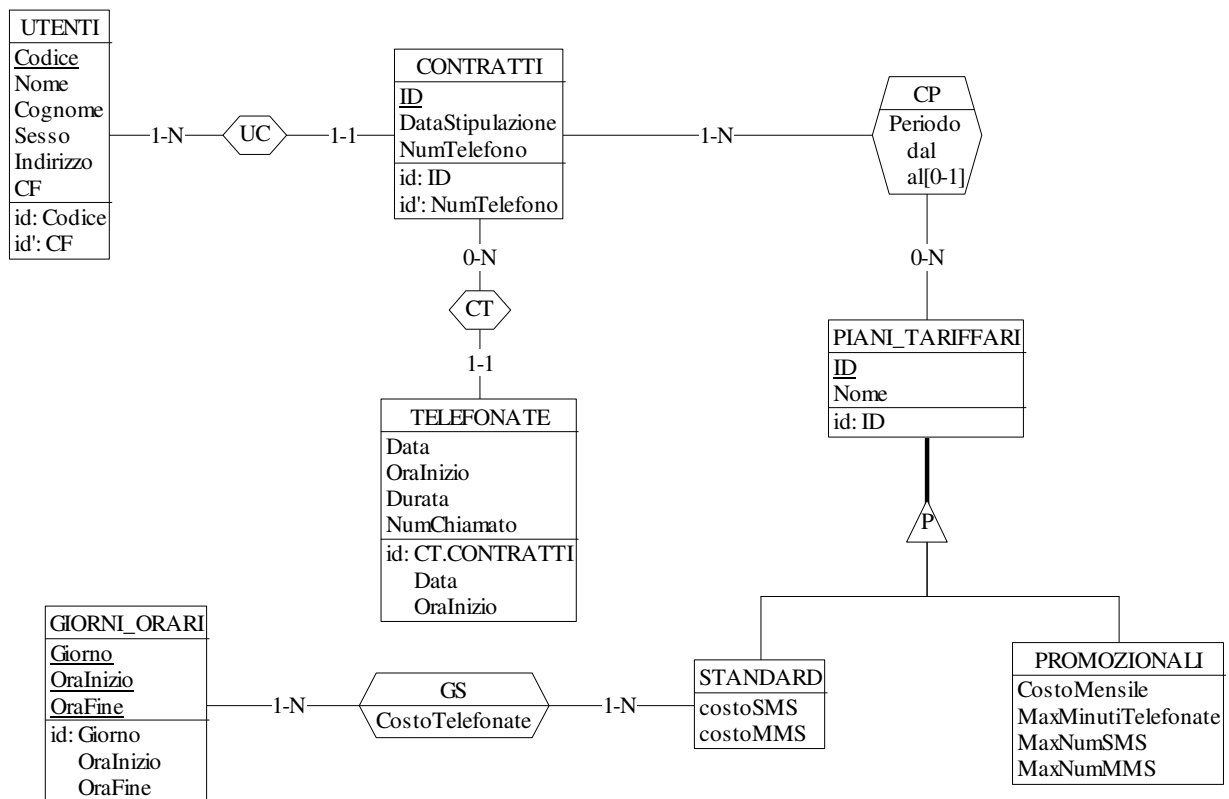
-- La c.t.e. calcola i CFU che ogni studente ha acquisito.
-- E' poi sufficiente raggruppare per fascia (NumCFU/50) per ottenere
-- il risultato
```

3) Progettazione concettuale (6 punti)

La DUE+UNO è una società telefonica che offre contratti per la telefonia mobile. Ogni contratto è caratterizzato dalla data di stipulazione e dal numero di telefono abbinato, ed è intestato ad un singolo utente (identificato da un codice e caratterizzato da nome, cognome, sesso, indirizzo e codice fiscale), ma un utente può essere titolare di più contratti. Alla stipula del contratto viene anche stabilito il relativo piano tariffario iniziale, che può essere o standard o promozionale.

Un piano standard si caratterizza per i costi di SMS e MMS e per quelli delle telefonate (questi ultimi variano in funzione del giorno della settimana e della fascia oraria). Per i piani promozionali si paga un costo fisso mensile e si ha una durata massima per le conversazioni e un numero massimo di SMS e di MMS inviabili ogni mese. Se un utente cambia il piano tariffario di un suo contratto, il sistema mantiene il periodo in cui il piano è stato attivo per quel contratto.

Il database della DUE+UNO mantiene anche informazioni su tutte le chiamate telefoniche effettuate dai propri clienti, registrando per ogni chiamata la data, l'orario di inizio, la durata e il numero chiamato.

**Commenti:**

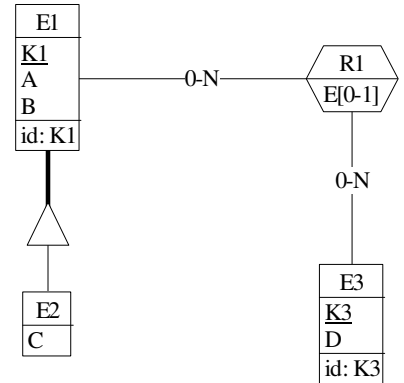
- Premesso che l'esercizio non presenta particolari difficoltà, l'unica osservazione degna di nota riguarda l'identificatore di GIORNI_ORARI, che include anche l'attributo OraFine. Tale scelta permette di definire, per uno stesso giorno, fasce diverse che iniziano alla stessa ora (e che evidentemente si riferiscono a piani tariffari diversi), ad es. LUN 14-19 e LUN 14-24.
- Nell'associazione CP, per un dato contratto solo l'ultimo piano tariffario (quello attualmente attivo) ha valore nullo per l'attributo Periodo.al.

Sistemi Informativi T
17 giugno 2014
Risoluzione

• **Progettazione logica (6 punti totali)**

Dato lo schema concettuale in figura e considerando che:

- tutti gli attributi sono di tipo INT;
- le entità E1 ed E2 vengono tradotte insieme;
- l'attributo E di R1 ha un valore sempre definito se all'associazione partecipa un'istanza di E2;
- vale la dipendenza funzionale $A \rightarrow B$;



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**

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

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

```
CREATE TABLE R1 (
  K1 INT NOT NULL REFERENCES E1,
  K3 INT NOT NULL REFERENCES E3,
  E INT,
  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 tuple non corrette**, definiti in un file **TRIGGER.txt** e usando se necessario il simbolo '@' per terminare gli statement SQL (altrimenti ';')

```
-- Trigger che garantisce il rispetto del vincolo di cui al punto c)
CREATE TRIGGER PUNTO_C
BEFORE INSERT ON R1
REFERENCING NEW AS N
FOR EACH ROW
WHEN (N.E IS NULL AND EXISTS ( SELECT * FROM E1
                                WHERE N.K1 = E1.K1
                                AND   E1.TIPO = 2      ))
SIGNAL SQLSTATE '70001' ('Se la tupla referenziata da K1 e" di E2 il valore di E non deve essere null!');
```

```
-- Trigger che garantisce il rispetto della dipendenza funzionale da A a B
CREATE TRIGGER A_DETERMINA_B
BEFORE INSERT ON E1
REFERENCING NEW AS N
FOR EACH ROW
WHEN (EXISTS ( SELECT * FROM E1
                WHERE N.A = E1.A
                AND N.B <> E1.B )
SIGNAL SQLSTATE '70002' ('La tupla viola la dipendenza funzionale da A a B!');
```