

Tempo a disposizione: 2:30 ore

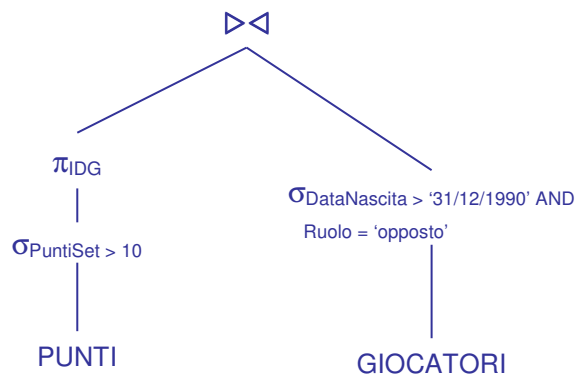
1) Algebra relazionale (3 punti totali):

Date le seguenti relazioni:

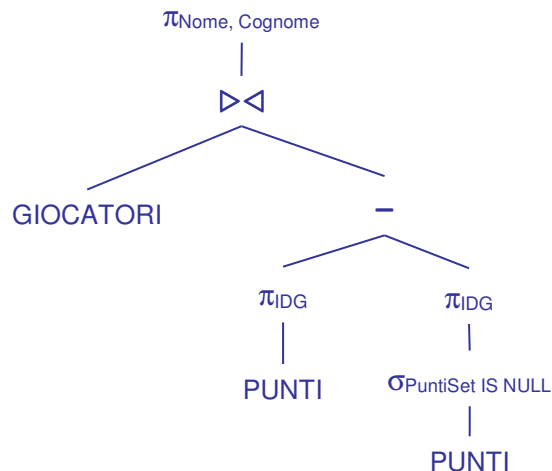
```
GIOCATORI (IDG, Nome, Cognome, DataNascita, Ruolo);
PUNTI (IDG, Partita, Set, PuntiSet*),
      IDG REFERENCES GIOCATORI;
--
-- Ogni Partita (di pallavolo) si compone di più Set (da 3 a 5).
-- PuntiSet è di tipo INT: è nullo se il giocatore in quel Set non ha
-- giocato, mentre vale 0 se ha giocato ma senza far punti.
-- Un giocatore può, per vari motivi, non prender parte a una partita.
-- In questo caso la relativa coppia (IDG, Partita) non compare nel DB.
```

si scrivano in algebra relazionale le seguenti interrogazioni:

- 1.1) [1 p.]** I dati dei giocatori nati dopo il 31/12/1990 che giocano nel ruolo di 'opposto' e che in almeno un set hanno fatto più di 10 punti



- 1.2) [2 p.]** Nome e cognome dei giocatori che hanno giocato tutti i set di tutte le partite cui han preso parte



Sostituendo GIOCATORI a PUNTI nell'operando sinistro della differenza si restituiscono anche i giocatori che non hanno preso parte a nessuna partita (non esplicitamente richiesto)

Sistemi Informativi T
21 gennaio 2019
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 giocatore, fornire identificativo, nome, cognome e la media dei set giocati nelle partite in cui ha preso parte

```
SELECT    G.IDG, G.Nome, G.Cognome,
          CAST(COUNT(PuntiSet) AS DEC(6,2))/COUNT(DISTINCT Partita)
          AS MEDIA_SET_GIOCATI
FROM      GIOCATORI G, PUNTI P
WHERE     G.IDG = P.IDG
GROUP BY  G.IDG,G.Nome,G.Cognome

-- COUNT(PuntiSet) restituisce il numero di set giocati, mentre
-- COUNT(DISTINCT Partita) conta le partite in cui un dato giocatore era
-- presente
```

- 2.2) [3 p.]** La partita in cui la differenza tra i punti realizzati da due giocatori è stata massima

```
WITH
PUNTI_PARTITA (Partita, IDG, TotPunti) AS (
    SELECT    P.Partita,P.IDG,SUM(P.PuntiSet)
    FROM      PUNTI P
    GROUP BY  P.Partita, P.IDG
),

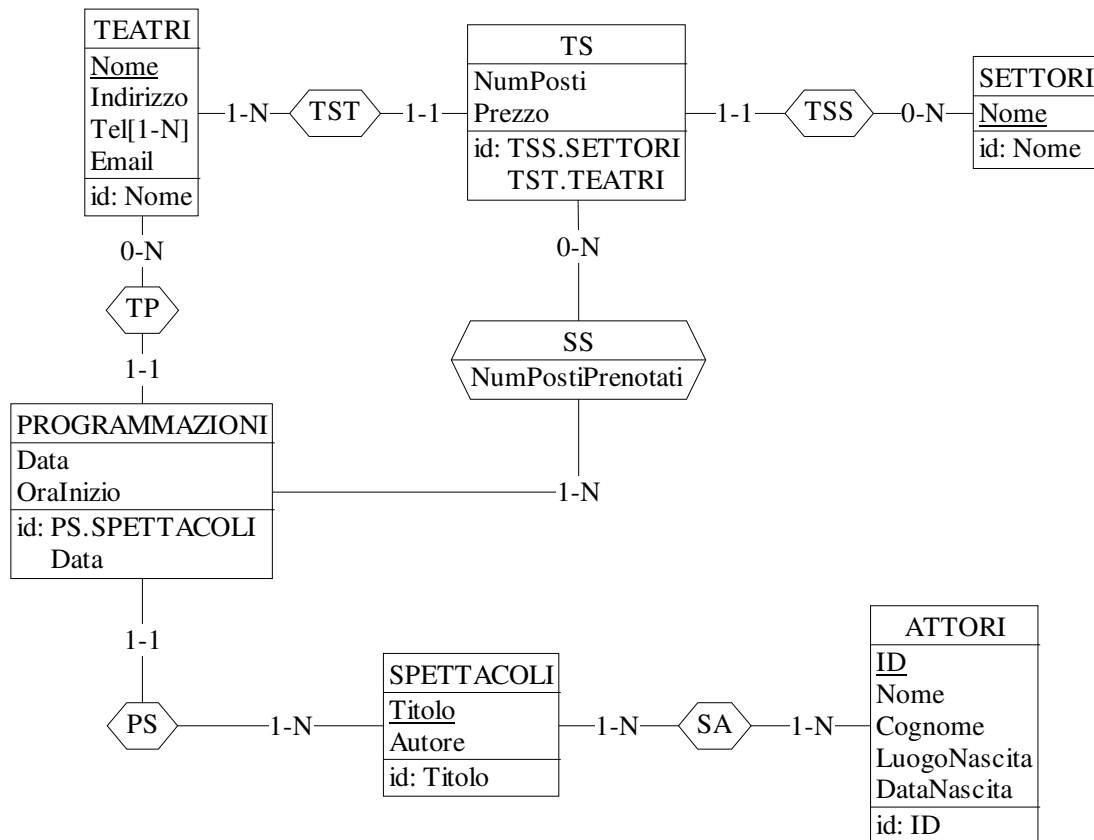
DIFF_PUNTI (Partita,Diff) AS (
    SELECT    P.Partita, MAX(TotPunti)-MIN(TotPunti)
    FROM      PUNTI_PARTITA P
    GROUP BY  P.Partita
)

SELECT    D1.*
FROM      DIFF_PUNTI D1
WHERE     D1.Diff = ( SELECT MAX(D2.Diff)
                     FROM    DIFF_PUNTI D2 )

-- La prima c.t.e. calcola i punti realizzati da ogni giocatore in ogni
-- partita, la seconda la massima differenza di punti in ogni partita
```

3) Progettazione concettuale (6 punti)

La Regione Emilia Romagna vuole realizzare un sistema per la gestione integrata dei Teatri Comunali. Per ogni teatro, oltre al nome, indirizzo, recapiti telefonici ed email, si memorizzano informazioni sul numero di posti in ogni settore (i settori, quali 'platea', 'galleria', 'palchi', ecc., sono codificati a priori) e sul relativo prezzo (il prezzo in un settore può ovviamente variare da un teatro all'altro). Relativamente agli spettacoli, di cui si memorizzano il titolo, l'autore e gli attori protagonisti (di questi si mantengono anche luogo e data di nascita), per ogni teatro in cui quello spettacolo è programmato si mantiene l'elenco completo dei giorni di programmazione, fornendo per ogni giorno l'orario di inizio (in un giorno uno spettacolo si tiene una volta sola, e quindi in un solo teatro) e il numero di posti prenotati in ogni settore.



Commenti:

- L'entità PROGRAMMAZIONI ha come identificatore lo spettacolo e la Data, in quanto in una certa data lo stesso spettacolo si tiene al massimo una volta.
- L'entità TS è ottenuta per reificazione, al fine di considerare, per ogni teatro, i soli settori presenti in quel teatro.
- Il vincolo che ogni istanza di PROGRAMMAZIONI si riferisce a settori di uno stesso teatro, e che tale teatro è lo stesso individuato dall'associazione TP, non è esprimibile a livello di schema E/R.
- Non è parimenti esprimibile il vincolo che NumPostiPrenotati sia non maggiore del NumPosti del corrispondente settore.
- La cardinalità minima pari a 1 di diverse partecipazioni con cardinalità massima N si può rilassare a 0 per aumentare la flessibilità in fase di popolamento del DB.

4) 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;
- un'istanza di E3 non è mai associata, tramite R1, all'istanza di E2 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 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,
  CONSTRAINT E2 CHECK ((TIPO = 1 AND B IS NULL) OR (TIPO = 2 AND B IS NOT NULL)) );
```

```
CREATE TABLE E3 (
  K1R2  INT NOT NULL REFERENCES E1,
  C     INT NOT NULL,
  D     INT NOT NULL,
  PRIMARY KEY (K1R2,C) );
```

```
CREATE TABLE R1 (
  K1    INT NOT NULL REFERENCES E1,
  K1E3  INT NOT NULL,
  CE3   INT NOT NULL,
  E     INT NOT NULL,
  PRIMARY KEY (K1,K1E3,CE3),
  CONSTRAINT FK_E3 FOREIGN KEY (K1E3,CE3) REFERENCES E3(K1R2,C),
  CONSTRAINT PUNTO_C CHECK (K1 <> K1E3) );
```

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 R2 referenzi un'istanza di E2

```
CREATE TRIGGER R2_E2
BEFORE INSERT ON E3
REFERENCING NEW AS N
FOR EACH ROW
WHEN ( NOT EXISTS ( SELECT *
                    FROM   E1
                    WHERE N.K1R2 = E1.K1
                    AND    E1.TIPO = 2 ) )
SIGNAL SQLSTATE '70001' ('La tupla referenzia una tupla che non appartiene a E2!');
```

-- L'inserimento di una tupla in E3 deve prevedere l'inserimento di almeno una tupla in R1, per rispettare
 -- il vincolo min-card(E3,R1) = 1. E' quindi necessario definire una transazione.

