

# Sistemi Informativi L-B

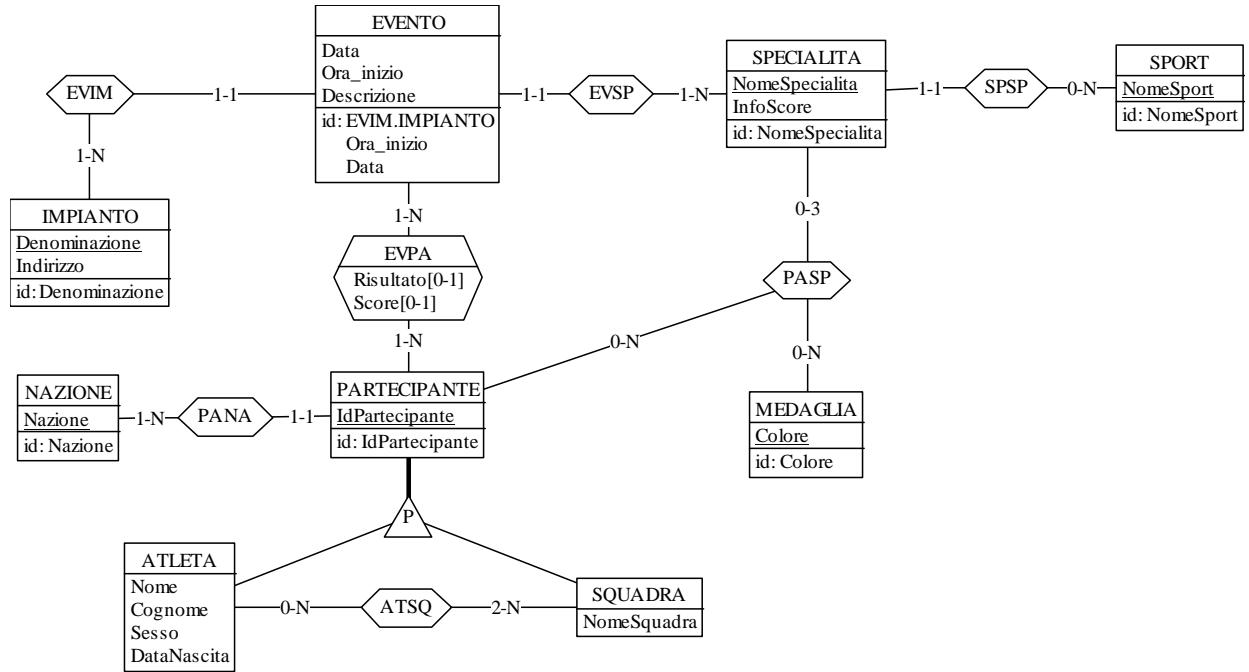
26 giugno 2008

Risoluzione

**Tempo a disposizione: 2 ore**

---

## 1) Progettazione concettuale (5 punti)



### Commenti:

- La soluzione proposta tiene conto del fatto che diverse informazioni non sono, all'atto della predisposizione del database, disponibili (vincitori delle medaglie e risultati dei singoli partecipanti). Questo si traduce in partecipazioni opzionali ad associazioni e in attributi con possibile valore nullo.
- Lo schema non distingue tra gare individuali e di squadra, e pertanto non modella esplicitamente il vincolo che ad un evento possono partecipare o solo atleti che gareggiano individualmente o solo squadre. In alternativa si sarebbero potuto introdurre due entità specializzazioni di EVENTO a cui associare, rispettivamente, ATLETA e SQUADRA.

## Sistemi Informativi L-B

26 giugno 2008

### Risoluzione

#### 2) Progettazione logica e normalizzazione (3 punti)

Si consideri lo schema R(ABCDEG) su cui sono definite le seguenti dipendenze funzionali:

$$F = \{ABC \rightarrow DE, BC \rightarrow D, DE \rightarrow A, CD \rightarrow BEG\}$$

Si normalizzi, preservando le dipendenze, lo schema R producendo schemi in 3NF e indicando per ciascuno di questi la chiave primaria e le eventuali altri chiavi individuate.

#### Risoluzione

Si inizia riscrivendo le dipendenze funzionali in forma semplice:

$$F = \{ABC \rightarrow D, ABC \rightarrow E, BC \rightarrow D, DE \rightarrow A, CD \rightarrow B, CD \rightarrow E, CD \rightarrow G\}$$

Calcolando le chiusure dei lati sinistri si ha:

$$\begin{aligned}ABC^+ &= ABCDEG \\BC^+ &= BCDEGA \\DE^+ &= DEA \\CD^+ &= CDBEGA\end{aligned}$$

Da cui si deriva che nelle dipendenze funzionali  $ABC \rightarrow D$  e  $ABC \rightarrow E$  l'attributo A è estraneo. Si ha quindi:

$$F' = \{BC \rightarrow E, BC \rightarrow D, DE \rightarrow A, CD \rightarrow B, CD \rightarrow E, CD \rightarrow G\}$$

La dipendenza  $BC \rightarrow E$  è ridondante, da cui:

$$F'' = \{BC \rightarrow D, DE \rightarrow A, CD \rightarrow B, CD \rightarrow E, CD \rightarrow G\}$$

Gli schemi che si ottengono da  $F''$  sono pertanto:

$$\begin{aligned}R1 &(\underline{BCD}) \\R2 &(\underline{DEA}) \\R3 &(\underline{CDBEG})\end{aligned}$$

Gli schemi R1 e R3 si fondono (vale sia  $BC \rightarrow CD$  che  $CD \rightarrow BC$ ) e si ottiene quindi come risultato finale:

$$\begin{aligned}\textbf{R13 (BCDEG)} &\quad \text{-- chiave alternativa CD} \\ \textbf{R2 (DEA)} &\end{aligned}$$

La chiave di R13 è anche chiave dello schema originario, quindi la decomposizione è senza perdita di dati (come si verifica facilmente osservando che il join naturale tra R13 e R2 è su DE, che è chiave di R2)

#### 3) Elaborazione delle interrogazioni (2 punti)

Con riferimento al metodo di join Nested Loops, si discuta come possono essere sfruttati eventuali indici presenti, avendo cura di distinguere tra quelli costruiti sulla relazione esterna R e quelli sulla relazione interna S. Si consideri il caso generale in cui sia su R che su S è presente (almeno) un predicato locale (oltre a quello di join)

I casi da considerare sono essenzialmente 3:

1. Indice utile per risolvere il predicato locale su R (relazione esterna)  
Si può usare l'indice per reperire le sole tuple di R che soddisfano il predicato e, per ognuna di queste, cercare i match su S
2. Indice utile per risolvere il predicato di join  
Per ogni tupla di R che soddisfa il predicato locale si sfrutta quest'indice per localizzare efficientemente i match su S
3. Indice utile per risolvere il predicato locale su S (relazione interna)  
In alternativa a 2., per ogni tupla di R che soddisfa il predicato locale si sfrutta quest'indice per reperire le sole tuple di S che soddisfano il predicato locale su S, e solo tra queste cercare i match per il predicato di join. Si noti che, a differenza del caso 2., ora l'indice supporta sempre lo stesso tipo di richiesta (ad esempio  $S.B > 40$ ). E' quindi ragionevole considerare che l'indice venga usato una sola volta e che le tuple di S che soddisfano il predicato siano salvate in una relazione temporanea in cui andare a cercare i match del join.

In conclusione, gli scenari che si possono considerare sono (nell'ipotesi che usare un indice convenga sempre rispetto alla scansione sequenziale) 1 e 2, oppure 1 e 3.