

Simulazione d'esame 3/06/2020

Si consideri il database **PremierLeague.sql**, contenente informazioni sul campionato di calcio inglese della stagione 2011/2012. Il database è strutturato secondo il diagramma ER illustrato nella pagina seguente. Si intende costruire un'applicazione JavaFX che permetta di interrogare tale base dati. L'applicazione dovrà svolgere le seguenti funzioni:

PUNTO 1

- Permettere all'utente di inserire, in un apposito campo di testo, un numero minimo x di *goal fatti* (campo **Goals**, tabella **actions**)
- Alla pressione del bottone "Crea Grafo", si crei un grafo semplice, pesato e orientato i cui nodi sono i giocatori che hanno *segnato almeno x goal in media a partita durante la stagione* (provare con valori < 1).
- Un arco collega due giocatori (tra quelli precedentemente selezionati) se appartengono a squadre diverse, e sono scesi in campo da "titolari" (campo **starts = 1**, tabella **actions**) in almeno una partita in cui si sono affrontati. Dati due giocatori, in particolare, l'arco deve essere orientato dal giocatore che ha giocato più minuti all'interno di queste partite (campo **TimePlayed**, tabella **actions**) verso il giocatore che ne ha giocati di meno. Il peso dell'arco, sempre ≥ 0 , rappresenta la differenza dei minuti giocati (Δ) dai due giocatori all'interno delle partite in cui si sono affrontati.
- Premendo il bottone "Top-Player", si trovi il giocatore che abbia "battuto", in termini di minuti giocati, il numero maggiore di avversari. Si stampino, contestualmente, gli avversari battuti ordinati in modo decrescente di Δ .

PUNTO 2

- Permettere all'utente di inserire un numero di giocatori k (**Suggerimento**: usare valori di K piccoli, < 5).
- Alla pressione del bottone "Dream-Team", si individui, tramite un algoritmo ricorsivo, un insieme di k giocatori (il "dream-team") che massimizzi il *grado di titolarità* del team. Il *grado di titolarità* di ogni singolo giocatore, in particolare, è dato dalla differenza del peso dei suoi archi uscenti (i minuti che ha giocato in più dei suoi avversari) con il peso degli archi entranti (i minuti che ha giocato in meno).
N.B.: i giocatori che entrano a far parte del dream-team possono far parte di squadre diverse.
- Quando un giocatore entra a far parte del dream-team, tutti i giocatori che quest'ultimo ha "battuto", in termini di minuti giocati, durante la stagione non possono più essere aggiunti alla squadra.
- Si stampi il "dream team" ottenuto e il suo *grado di titolarità*.

Nella realizzazione del codice, si lavori a partire dalle classi (Bean e DAO, FXML) e dal database contenuti nel progetto di base. È ovviamente permesso aggiungere o modificare classi e metodi.

Tutti i possibili errori di immissione, validazione dati, accesso al database, ed algoritmici devono essere gestiti, non sono ammesse eccezioni generate dal programma.

Le tabelle **Teams**, **Players** e **Matches** contengono rispettivamente informazioni su squadre, giocatori e partite. Ogni riga della tabella **Actions**, invece, contiene le informazioni statistiche di un determinato giocatore in una determinata partita (minuti giocati, goal fatti, ecc.)

