

Esame del 30/06/2023 – Turno B

Si consideri il database contenuto nel file **lahmansbaseballdb_small.sql**, presente nella cartella “db” del progetto.

La struttura del database consiste di quattro tabelle principali: *teams*: lista di tutte le squadre iscritte al campionato per ogni anno dal 1871 al 2019; *people*: lista che raccoglie i dettagli dei giocatori professionisti; *appearances*: lista delle partecipazioni dei giocatori al campionato con una (o più) squadre (si noti che non è escluso a priori che un giocatore possa aver cambiato squadra durante l’anno di campionato); *salaries*: lista degli ingaggi del singolo giocatore per uno specifico anno. Il diagramma ER del database è fornito nella cartella “db” ed è riportato nella pagina seguente.

Si intende costruire un’applicazione JavaFX che permetta di svolgere le seguenti funzioni:

PUNTO 1

- Permettere all’utente di selezionare da un apposito menu a tendina il nome di una squadra (considerando tutte le squadre che hanno giocato in qualsiasi anno).
- Alla pressione del bottone “Crea Grafo”, creare un grafo semplice, **non orientato**, **pesato**, in cui i vertici siano gli **anni** in cui tale squadra ha giocato.
- Il grafo dovrà essere **completo**, e per ciascuna coppia di anni y_1, y_2 , il **peso** dell’arco che li connette dovrà essere pari alla **differenza in valore assoluto** del **salario cumulativo** delle due squadre (somma dei salari di tutti i giocatori che anno giocato nella squadra nell’anno y_1 e nell’anno y_2).

NOTA: nella scelta dell’algoritmo, tenere presente che l’elenco dei giocatori di ciascun anno è un’informazione necessaria per lo svolgimento del Punto 2.

- Permettere all’utente di selezionare, attraverso un secondo menu a tendina, uno degli anni esistenti nel grafo, ed alla pressione del bottone “Dettagli” stampare per tale anno l’elenco degli anni adiacenti, ed il peso degli archi corrispondenti, in ordine decrescente di peso.

PUNTO 2

Partendo dalle informazioni calcolate nel punto precedente, si implementi una simulazione ad eventi discreti, per poter valutare il fenomeno della *fanbase retention*, ossia dell’evoluzione dei tifosi relativi ai diversi giocatori della squadra.

La simulazione dovrà considerare solo la squadra selezionata nel punto 1.a. e dovrà considerare come tempo iniziale l’anno selezionato nel punto 1.d. L’utente dovrà altresì specificare il numero totale di tifosi di cui simulare il comportamento.

Si assuma che ciascuno tifoso possa essere, in ogni anno, “fan” di un solo giocatore. Nel primo anno, i tifosi sono divisi in parti uguali tra i giocatori presenti in quell’anno.

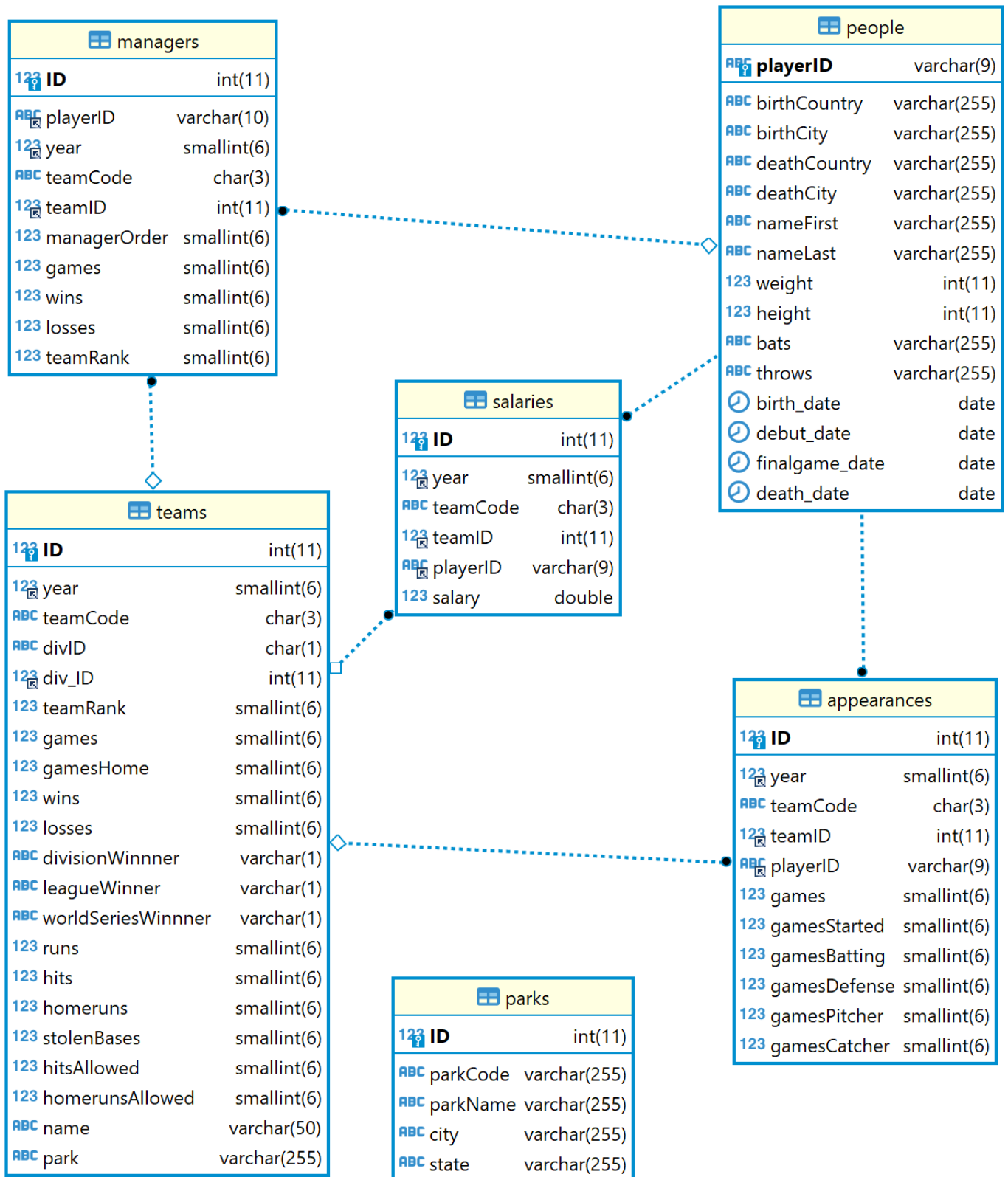
La simulazione deve considerare, in ordine, tutti gli anni successivi a quello di partenza in cui la squadra ha giocato. In ogni anno, i tifosi possono modificare la propria preferenza, secondo le regole seguenti:

- se il giocatore di cui sono tifosi gioca ancora nella squadra l'anno successivo: il 90% dei tifosi continuerà ad essere "fan" dello stesso giocatore, mentre il 10% modificherà la propria preferenza, scegliendo casualmente uno dei giocatori dell'anno successivo (diverso da quello precedente)
- se il giocatore preferito non gioca più nella squadra nell'anno successivo: il 10% dei tifosi smette di supportare la squadra (esce dalla simulazione), i rimanenti sceglieranno un giocatore diverso. Nella scelta del nuovo giocatore preferito, il 75% delle probabilità ricadono sui giocatori "nuovi" (che non erano presenti nell'anno precedente), il 25% sui giocatori che erano già presenti.

Alla fine della simulazione, stampare la statistica del numero di tifosi per ciascun giocatore, nell'anno finale della simulazione, ordinata alfabeticamente per nome del giocatore. Si stampi inoltre il numero di tifosi "persi" in totale.

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. Nelle pagine seguenti, sono disponibili due esempi di risultati per controllare la propria soluzione.



ESEMPI DI RISULTATI PER CONTROLLARE LA PROPRIA SOLUZIONE:

2023-05-30-esame

Esame 30/06/2023 - Turno B

Squadra Crea Grafo

Anno Dettagli

Tifosi Simula Tifosi

Grafo creato
#Vertici: 86
#Vertici: 3655

Dettagli per l'anno scelto:
2006<->anno: 2016; peso: 89277874
2006<->anno: 1882; peso: 72585582
2006<->anno: 1883; peso: 72585582
2006<->anno: 1884; peso: 72585582
2006<->anno: 1885; peso: 72585582
2006<->anno: 1886; peso: 72585582
2006<->anno: 1887; peso: 72585582
2006<->anno: 1888; peso: 72585582
2006<->anno: 1889; peso: 72585582
2006<->anno: 1890; peso: 72585582
2006<->anno: 1891; peso: 72585582
2006<->anno: 1892; peso: 72585582
2006<->anno: 1893; peso: 72585582
2006<->anno: 1894; peso: 72585582
2006<->anno: 1895; peso: 72585582
2006<->anno: 1896; peso: 72585582
2006<->anno: 1897; peso: 72585582
2006<->anno: 1898; peso: 72585582
2006<->anno: 1899; peso: 72585582
2006<->anno: 1901; peso: 72585582

2023-05-30-esame

Esame 30/06/2023 - Turno B

Squadra Crea Grafo

Anno Dettagli

Tifosi Simula Tifosi

Grafo creato
#Vertici: 129
#Vertici: 8256

Dettagli per l'anno scelto:
1993<->anno: 2016; peso: 78956366
1993<->anno: 2015; peso: 64070032
1993<->anno: 2014; peso: 52355533
1993<->anno: 2013; peso: 52239533
1993<->anno: 2012; peso: 38129532
1993<->anno: 2001; peso: 32938366
1993<->anno: 2003; peso: 29989962
1993<->anno: 1891; peso: 24822467
1993<->anno: 1892; peso: 24822467
1993<->anno: 1893; peso: 24822467
1993<->anno: 1894; peso: 24822467
1993<->anno: 1895; peso: 24822467
1993<->anno: 1896; peso: 24822467
1993<->anno: 1897; peso: 24822467
1993<->anno: 1898; peso: 24822467
1993<->anno: 1899; peso: 24822467
1993<->anno: 1900; peso: 24822467
1993<->anno: 1901; peso: 24822467
1993<->anno: 1902; peso: 24822467
1993<->anno: 1903; peso: 24822467