Fsame 07/11/2023

## Esame del 07/11/2023 – appello riservato

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 che svolga le seguenti funzioni:

## PUNTO 1

- a. Permettere all'utente di selezionare da un apposito menu a tendina il valore di un anno di campionato, tra quelli disponibili nel database (teams.year), a partire dal 1980 (gli anni precedenti non devono comparire).
- b. Non appena viene selezionato l'anno (intercettando l'evento onAction), si dovrà stampare il numero di squadre (teams) che ha giocato in tale anno, e l'elenco delle rispettive sigle, nella prima area di testo occorre & fill old - solitore (txtSquadre). stesso aggiornare il contenuto della tendina "Squadre".
- c. Alla pressione del bottone "Crea Grafo", occorre creare un grafo completo, non ordinato, pesato, in cui i vertici siano le squadre di cui al punto precedente, e gli archi colleghino tutte le coppie distinte di squadre.
- d. Il peso di ciascun arco del grafo deve corrispondere alla somma dei salari dei giocatori delle due squadre nell'anno considerato.

n. 59

- Nota: potrebbe essere conveniente calcolare e memorizzare la somma dei salari di ciascuna squadra.
- e. Permettere all'utente di selezionare, attraverso un secondo menu a tendina, una delle squadre esistenti nel grafo, ed alla pressione del bottone "Dettagli" stampare per tale squadra l'elenco delle squadre adiacenti, ed il peso degli archi corrispondenti, in ordine decrescente di peso. Tali informazioni dovranno essere stampate nella seconda area di testo (txtResult).

## PUNTO 2

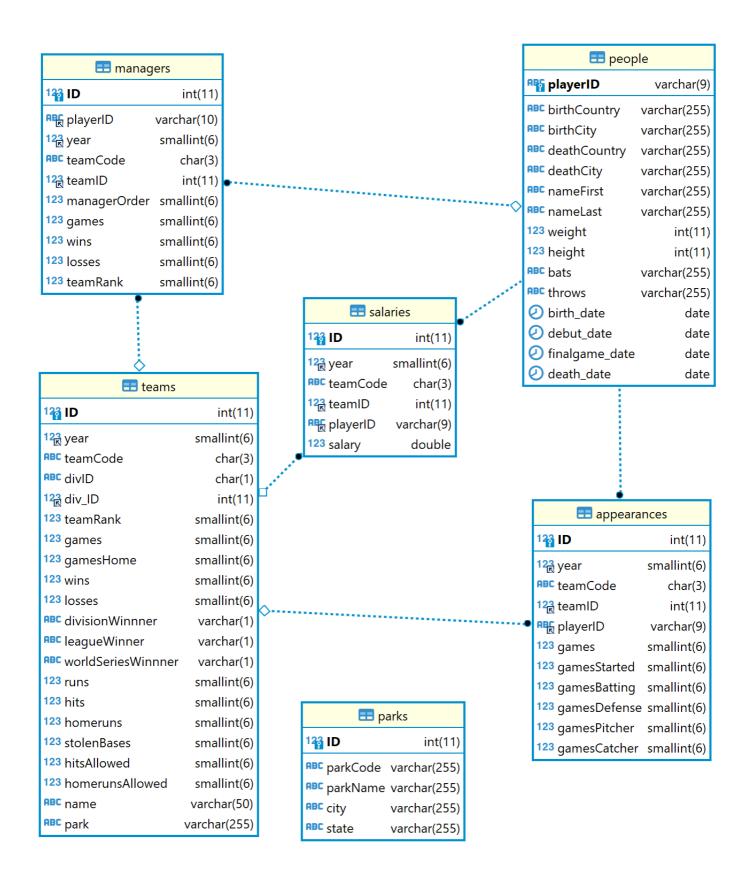
Partendo dal grafo calcolato nel punto precedente, alla pressione del bottone "Percorso" si implementi una procedura ricorsiva che calcoli un *percorso di peso massimo* avente le seguenti caratteristiche:

- Il punto di partenza è il vertice selezionato al punto 1.e.
- Ogni vertice può comparire una sola volta
- Il peso degli archi nel percorso deve essere strettamente decrescente.

Al termine della ricerca, si stampi l'elenco dei vertici ed i pesi degli archi incontrati, oltre al peso totale del percorso ottimo.

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:**

