

Esame del 02/11/2022 – *appello riservato*

Si consideri il database **iTunes.sql**, presente nella cartella “database” del progetto base e tratto dalla piattaforma iTunes di Apple. Esso contiene informazioni su artisti (Artist), album musicali (Album) e canzoni (Track), ed è stato estratto a partire dai dati di un utente reale. Il diagramma ER del database è illustrato nella pagina seguente.

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

PUNTO 1

- Permettere all’utente di selezionare, dall’apposita tendina uno dei generi musicali presenti nella tabella *Genre* (visualizzati in ordine alfabetico).
- In seguito, permettere all’utente di inserire il valore *minimo* ed il valore *massimo* della durata dei brani (campo *milliseconds* della tabella *Track*), inserito in **secondi**. Tali valori dovranno essere coerenti con la durata dei brani effettivamente presenti nel genere selezionato (in altre parole, **il minimo selezionato dell’utente deve essere maggiore o uguale alla durata minima dei brani presenti in quel genere**).
- Alla pressione del bottone “Crea grafo”, costruire un grafo semplice, non orientato e non pesato, **i cui vertici siano le Track che soddisfino i criteri di genere e di durata sopra specificati**.
- Ogni coppia di vertici è collegata da un arco, se e solo se il numero distinto di playlist in cui ciascuno dei brani corrispondenti ai due vertici è uguale.
- Al termine della costruzione del grafo, determinarne le componenti connesse e, per ciascuna di esse, stampare il numero di vertici presenti, ed **il numero di playlist distinte che corrispondono agli archi di tale componente**.

The screenshot shows a JavaFX application window with the title bar '2022-11-02 Esame 02/11/2022'. The window has a light gray background. At the top left, there is a 'Genere' label followed by a dropdown menu. To the right of the dropdown is a blue button labeled 'Crea Grafo'. Below the 'Genere' label are two input fields labeled 'Min' and 'Max'. Below these is a label 'dTOT' followed by another input field. At the bottom right, there is a button labeled 'La mia playlist'.

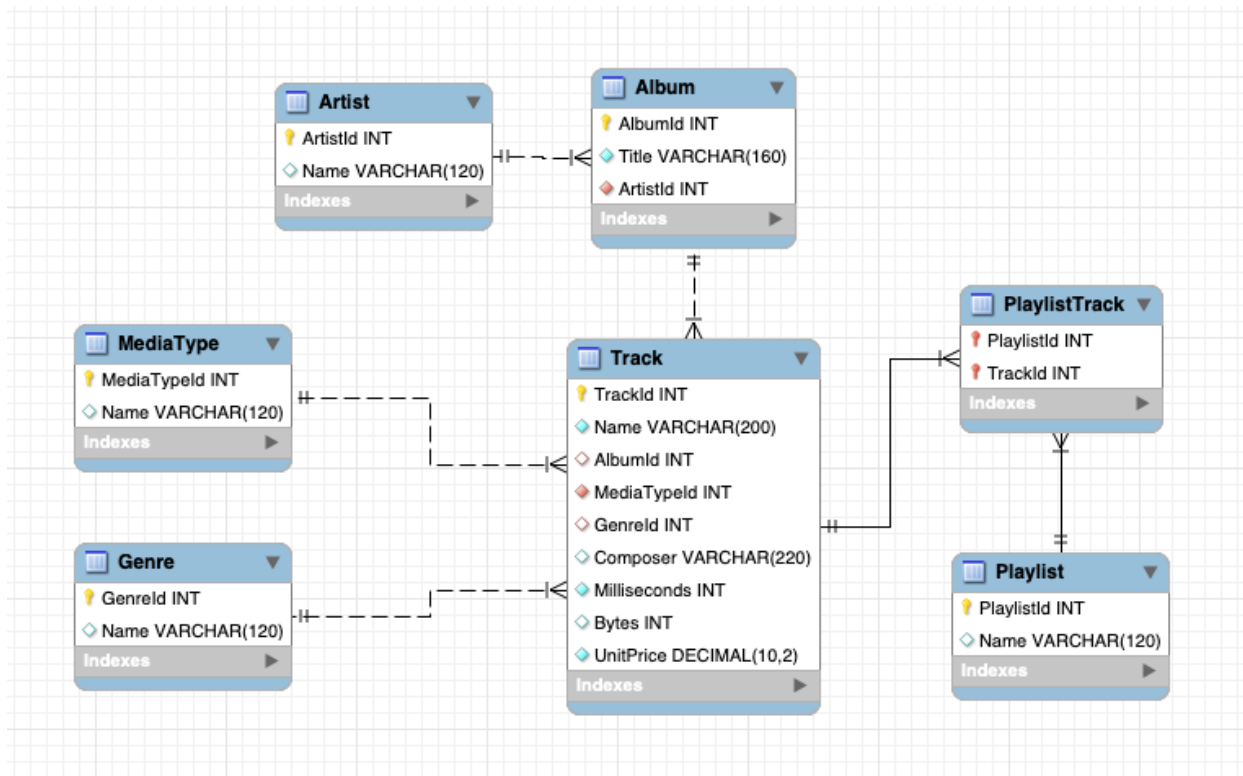
PUNTO 2

Permettere all’utente di inserire una durata complessiva *dTOT*, espressa in **minuti**. Alla pressione del bottone “La mia playlist”, utilizzare un algoritmo ricorsivo per calcolare l’insieme costituito dal **maggior possibile numero di brani, appartenenti alla componente connessa più numerosa**, che complessivamente abbia una durata minore o uguale a *dTOT*.

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.

Le tabelle **Artist**, **Album** e **Track** contengono rispettivamente informazioni su artisti, album musicali e canzoni contenute. Ogni canzone è collegata a un genere (tabella **Genre**) ed è memorizzata con uno specifico formato (tabella **MediaType**). La tabella **Playlist** contiene informazioni sulle playlist salvate dall'utente. La tabella **PlaylistTrack** modella la relazione molti a molti tra playlist e canzoni.



ESEMPI DI RISULTATI PER CONTROLLARE LA PROPRIA SOLUZIONE:

2022-11-02
Esame 02/11/2022

Genere:

Min:

Max:

dTOT:

Grafo creato!
Vertici : 472
Archi : 58081

Componente con 172 vertici, inseriti in 3 playlist
Componente con 295 vertici, inseriti in 2 playlist
Componente con 5 vertici, inseriti in 4 playlist

2022-11-02
Esame 02/11/2022

Genere:

Min:

Max:

dTOT:

Grafo creato!
Vertici : 60
Archi : 991

Componente con 41 vertici, inseriti in 2 playlist
Componente con 19 vertici, inseriti in 3 playlist

2022-11-02
Esame 02/11/2022

Genere:

Min:

Max:

dTOT:

Grafo creato!
Vertici : 340
Archi : 28418

Componente con 174 vertici, inseriti in 2 playlist
Componente con 164 vertici, inseriti in 3 playlist
Componente con 2 vertici, inseriti in 4 playlist