

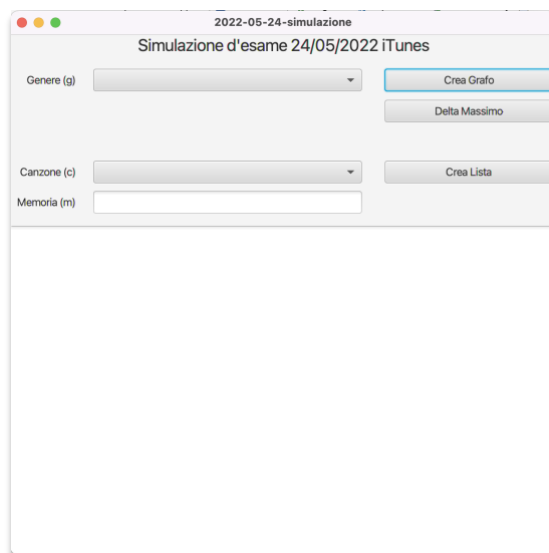
# Simulazione d'esame 24/05/2022

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, un genere  $g$  (tabella *Genre*).
- Alla pressione del bottone “Crea Grafo”, si crei un grafo semplice, non orientato e pesato, i cui vertici sono tutte le canzoni (tabella *Track*) di genere  $g$ .
- Due canzoni sono collegate tra loro se condividono lo stesso formato di file (*MediaType*). Il peso dell'arco, sempre positivo, rappresenta il valore assoluto della differenza di durata tra le due canzoni ( $\Delta$  **durata**), espressa in millisecondi.
- Alla pressione del bottone “Delta Massimo” Trovare nel grafo e stampare a video la coppia di canzoni collegate che abbia  $\Delta$  **durata** massimo, nel formato *titolo canzone1, titolo canzone 2,  $\Delta$  durata* (vedere screenshot nelle pagine seguenti). Nel caso in cui ci sia più di una coppia che abbia  $\Delta$  **durata** massimo, stamparle tutte.



## PUNTO 2

Un utente possiede un vecchio lettore MP3 su cui è interessato il maggior numero possibile di canzoni scaricate da iTunes. Il lettore MP3 ha una capacità di memoria limitata, e non è in grado di riprodurre file di formato diverso. L'utente ha una canzone preferita, che vuole sicuramente caricare sul lettore MP3.

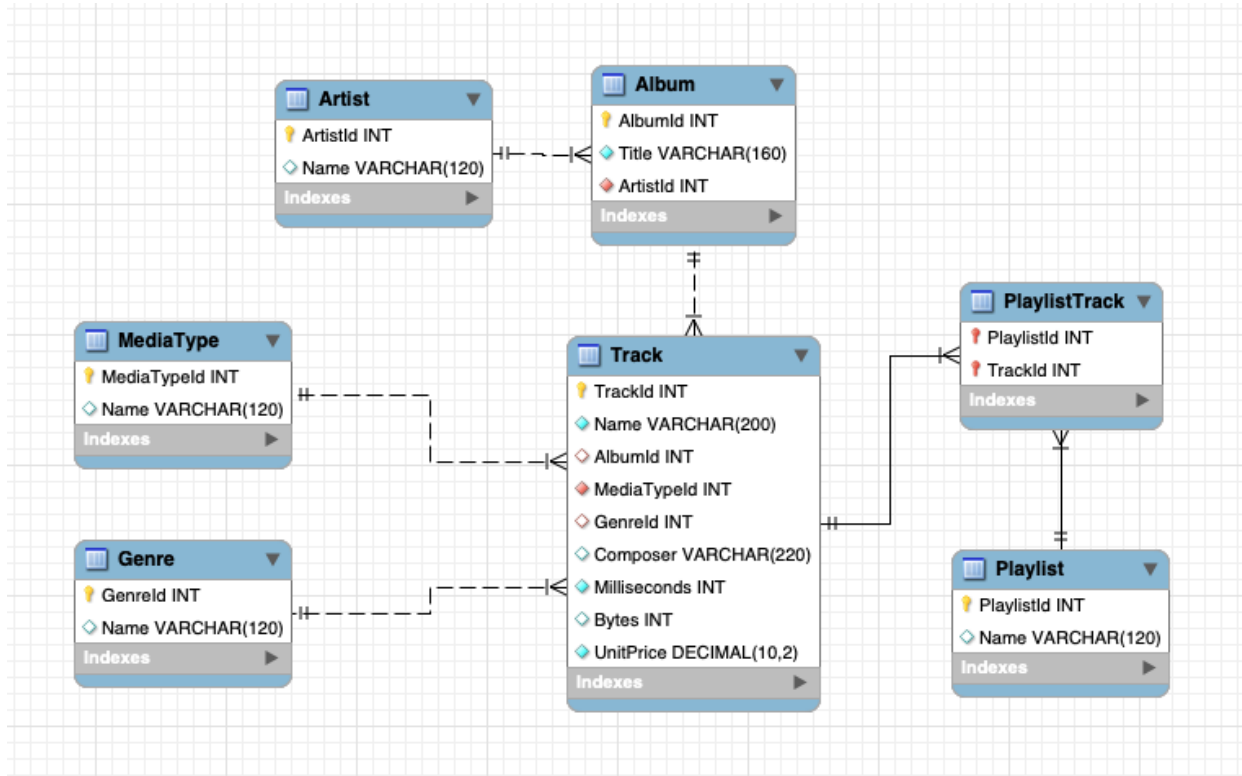
Permettere all'utente di selezionare la canzone preferita  $c$  (tra quelle presenti nel grafo), e di inserire una capacità di memoria massima  $m$ , espressa in numero bytes. Alla pressione del bottone “Crea Lista”, utilizzare un algoritmo ricorsivo per trovare l'insieme **più numeroso** di canzoni che abbia le seguenti caratteristiche:

- contenga la canzone  $c$ ;
- includa solamente canzoni che appartengano alla stessa componente connessa di  $c$ ;
- abbia una dimensione totale, definita come la somma dei *Bytes* di ogni canzone contenuta, non superiore alla capacità massima di memoria  $m$ ;

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:**

2022-05-24-simulazione

Simulazione d'esame 24/05/2022 iTunes

Genere (g)

Canzone (c)

Memoria (m)

Grafo creato!  
#VERTICI: 130  
#ARCHI: 8004

2022-05-24-simulazione

Simulazione d'esame 24/05/2022 iTunes

Genere (g)

Canzone (c)

Memoria (m)

COPPIA CANZONI DELTA MASSIMO:  
My Funny Valentine (Live) \*\*\* Outra Vez -> 781009

2022-05-24-simulazione

Simulazione d'esame 24/05/2022 iTunes

Genere (g)

Canzone (c)

Memoria (m)

Grafo creato!  
#VERTICI: 12  
#ARCHI: 66

2022-05-24-simulazione

Simulazione d'esame 24/05/2022 iTunes

Genere (g)

Canzone (c)

Memoria (m)

COPPIA CANZONI DELTA MASSIMO:  
Good Golly Miss Molly \*\*\* Slow Down -> 56999