

Prova d'esame del 02/07/2018 – Turno C

Si consideri il database “extflightdelays”, contenente informazioni su aeroporti, compagnie aeree, orari di partenza e di arrivo dei voli effettuati negli Stati Uniti durante il 2015. Il database (disponibile su Kaggle all'indirizzo: <https://www.kaggle.com/usdot/flight-delays/data>) è strutturato secondo il diagramma ER della pagina seguente.

Si intende costruire un'applicazione JavaFX che permetta di interrogare tale base dati, e calcolare informazioni a proposito dei voli tra i diversi aeroporti. L'applicazione dovrà svolgere le seguenti funzioni:

PUNTO 1

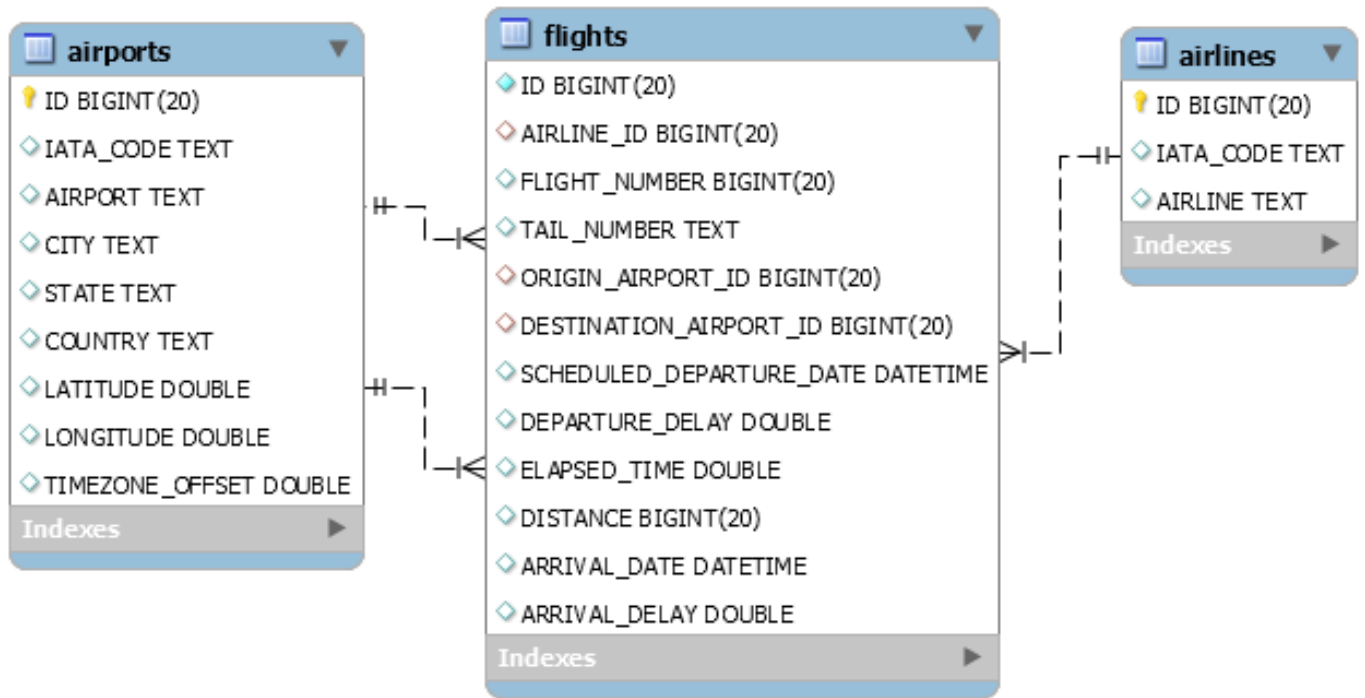
- Permettere all'utente di inserire un numero di compagnie aeree minimo x , e di selezionare il bottone “Analizza aeroporti”.
- Alla pressione del bottone, creare un grafo che rappresenti gli aeroporti su cui operano più compagnie.
Il grafo deve essere semplice, non orientato e pesato, i vertici devono rappresentare gli aeroporti su cui operano almeno x compagnie aeree (in arrivo o in partenza), e gli archi devono rappresentare le rotte tra gli aeroporti collegati tra di loro da almeno un volo. Il peso dell'arco deve rappresentare il numero totale di voli tra i due aeroporti (poiché il grafo non è orientato, considerare tutti i voli in entrambe le direzioni: **A->B** e **B->A**).
- Permettere all'utente di selezionare, da un menu a tendina, uno degli aeroporti presenti nel grafo (**a1**).
- Alla pressione del bottone “Aeroporti connessi”, stampare l'elenco degli aeroporti adiacenti a quello selezionato, in ordine decrescente di numero totale di voli.

PUNTO 2

- Permettere all'utente di selezionare da un menu a tendina un aeroporto di destinazione **a2**, e un numero massimo di tratte t che è disposto a percorrere.
- Alla pressione del bottone “Cerca itinerario”, sviluppare un algoritmo ricorsivo agente sul grafo creato al punto 1 per cercare l'itinerario di viaggio tra **a1** e **a2** che massimizzi il numero totale di voli per ciascuna delle tratte del percorso selezionato (in altre parole, il percorso che massimizzi la somma dei pesi degli archi attraversati), utilizzando al massimo t tratte.
- Al termine della ricerca, il programma dovrà stampare l'itinerario, indicando gli aeroporti visitati, e il numero totale di voli disponibili sul percorso.

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.



La tabella *airports* riporta tutti gli aeroporti presenti sul territorio statunitense, la tabella *airlines* tutte le compagnie aeree operanti, mentre la tabella *flights* contiene informazioni sui voli effettuati da una specifica compagnia aerea tra una coppia di aeroporti, riportando le date di partenza, di arrivo ed eventuali ritardi.