

Simulazione del 13/06/2018

Si consideri il database “flightdelays”, contenente informazioni su aeroporti, compagnie aeree ed orari di partenza e di arrivo dei voli effettuati negli Stati Uniti durante il 2015. I dati sono stati estratti dal database disponibile su Kaggle all’indirizzo: <https://www.kaggle.com/usdot/flight-delays/data>

flights	airports	airlines
ID BIGINT(20)	ID TEXT	ID TEXT
AIRLINE TEXT	AIRPORT TEXT	AIRLINE TEXT
FLIGHT_NUMBER BIGINT(20)	CITY TEXT	
TAIL_NUMBER TEXT	STATE TEXT	
ORIGIN_AIRPORT_ID TEXT	COUNTRY TEXT	
DESTINATION_AIRPORT_ID TEXT	LATITUDE DOUBLE	
SCHEDULED_DEP_DATE DATETIME	LONGITUDE DOUBLE	
DEPARTURE_DELAY DOUBLE		
AIR_TIME DOUBLE		
DISTANCE BIGINT(20)		
ARRIVAL_DATE DATETIME		
ARRIVAL_DELAY DOUBLE		
Indexes		

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.

Si intende costruire un’applicazione JavaFX che permetta di interrogare tale base dati. L’applicazione dovrà svolgere le seguenti funzioni:

Flight Delays - Simulazione 2018-06-13

Linea aerea Carica voli

Numero passeggeri

Numero voli Simula

PUNTO 1

- a. L'utente seleziona una linea aerea tra quelle disponibili, utilizzando l'apposito menù a tendina.
- b. Facendo click sul pulsante *"Carica voli"*, costruire un grafo semplice, orientato e pesato i cui nodi siano gli aeroporti. Un arco collega due aeroporti solo esiste almeno un volo della compagnia aerea selezionata tra gli aeroporti considerati.
- c. Il peso dell'arco è dato dal rapporto tra la media dei ritardi su quella tratta (*"ARRIVAL_DELAY"*) e la distanza geografica tra gli aeroporti (utilizzare la libreria *SimpleLatLng* per calcolare la distanza tra le coordinate degli aeroporti).
- d. Stampare le 10 peggiori rotte, ossia i 10 archi di peso massimo.

PUNTO 2

- a. L'utente inserisce nelle apposite caselle di testo un numero K di passeggeri ed un numero V di voli.
- b. Facendo click sul pulsante *"Simula"*, simulare lo spostamento di K passeggeri con comportamento casuale. Inizialmente, posizionare K passeggeri in modo casuale tra gli aeroporti disponibili. In seguito, ciascun passeggero prende il primo volo del 2015 in partenza (*"SCHEDULED_DEPARTURE_DATE"*) da quell'aeroporto.
- c. Il passeggero, non appena giunto a destinazione (*"ARRIVAL_DATE"*), sceglierà il successivo **primo volo disponibile** in partenza da tale aeroporto. Ipotizzare che non vi siano problemi di capienza e trascurare i tempi di imbarco. La simulazione termina dopo che **ciascun passeggero ha preso V voli**, oppure se non ci sono più voli in partenza dall'aeroporto considerato.
- d. Al termine della simulazione, stampare il ritardo complessivo (*"ARRIVAL_DELAY"*) accumulato da ciascun passeggero.

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.