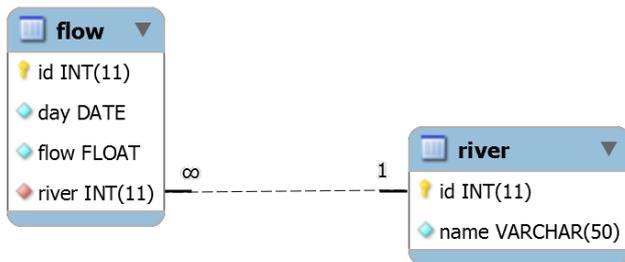


Prova d'esame del 10/07/2015 – Turno A

Si consideri il data-set “**Rivers**”, ottenuto combinando un sottoinsieme delle informazioni presenti nella *Time Series Data Library*¹ (TSDL). Tale data-set contiene la portata dei flussi di acqua di alcuni fiumi, misurati quotidianamente per diversi anni. Notare che i diversi fiumi sono stati misurati in intervalli di tempo indipendenti tra loro. Si noti anche che le portate dei fiumi sono molto diverse tra loro, di vari ordini di grandezza.

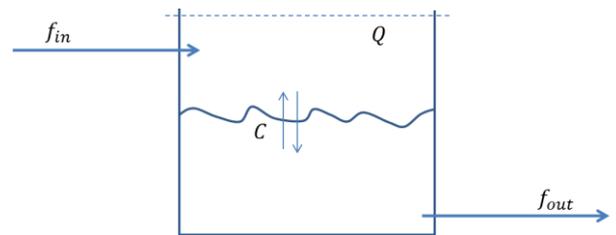
La struttura del database è la seguente:



La tabella ‘river’ contiene i fiumi considerati nel data set, ciascuno rappresentato da un identificatore numerico ‘id’ e da una stringa descrittiva ‘name’.

La tabella ‘flow’ contiene le misurazioni effettuate, nelle date specificate dal campo ‘day’ e per il fiume identificato dal campo (chiave esterna) ‘river’. La colonna ‘flow’ riporta il flusso misurato nel giorno specificato, espresso in *metri cubi per secondo*. La chiave primaria ‘id’ è un semplice identificatore univoco della misurazione.

Si intende costruire un’applicazione JavaFX che permetta di interrogare tale base dati, e supportare un progettista che ha l’incarico di costruire la diga per la realizzazione di un bacino idrico che si inserisce nel flusso del fiume (v. schema a fianco).



L’applicazione dovrà svolgere le seguenti funzioni:

1. Permettere all’utente di selezionare, da una tendina, uno specifico fiume. Appena il fiume è selezionato, occorre riempire i campi dell’interfaccia che riportano, rispettivamente: la data della prima misurazione disponibile per tale fiume, la data dell’ultima misurazione disponibile per tale fiume, il numero totale di misurazioni, il valore medio del flusso misurato per tale fiume (detto f_{med}).
2. Permettere all’utente di simulare l’effetto di un bacino idrico della capienza totale Q (espressa in metri cubi). La simulazione dovrà aggiornare, giorno per giorno, la quantità di acqua presente nel bacino (detta C), in funzione dei dati sul flusso in ingresso (detto f_{in}) presente nel data set. Il calcolo deve tenere conto delle seguenti regole:
 - a. L’utente può specificare il valore di Q inserendo un “fattore di scala” $k > 0$, dal quale si calcherà² $Q = k \cdot f_{med} \cdot 30 \text{ giorni}$ (in altre parole, k indica la frazione di un mese a flusso “medio” che il bacino potrà contenere prima di riempirsi).
 - b. L’occupazione iniziale del bacino è $C = Q/2$
 - c. Il bacino dovrà garantire un flusso in uscita (detto f_{out}) pari ad almeno $f_{out_{min}}$, ogni giorno dell’anno. Per gli scopi della simulazione, si assuma $f_{out_{min}} = 0.8 \cdot f_{med}$.
 - d. Quando $f_{in} > f_{out}$, il livello C nel bacino salirà di una quantità legata alla differenza tra i due flussi.
 - e. Non è possibile avere $C > Q$. In caso di flusso in ingresso eccessivo, questo deve essere scaricato in uscita (evento detto “tracimazione”).

¹ <https://datamarket.com/data/list/?q=interval:day%20provider:tSDL>, create da Rob Hyndman, Professore di Statistica alla Monash University, Australia

² Attenzione alle unità di misura, in quanto f_{med} è espresso in m^3 al secondo e non al giorno

- f. In ogni giorno dell'anno, c'è una probabilità pari al 5% di avere un flusso richiesto in uscita più elevato, pari a $10 \cdot f_{out_min}$, in quanto i campi devono essere irrigati.

Il simulatore dovrà determinare, in funzione di k :

- il numero di giorni in cui non si è potuta garantire l'erogazione minima;
- l'occupazione media C_{med} del bacino nel corso della simulazione.

The screenshot shows a Java Swing window titled "Rivers (Esame 10/07/2015, turno A)". The window contains the following elements:

- A dropdown menu labeled "Fiume" with the text "Scegli un fiume..." and a downward arrow.
- Four text input fields arranged horizontally, labeled "Prima data", "Ultima data", "N. misure", and "f_med".
- A text input field labeled "Fattore di scala (k)" and a "Simula" button.
- A large empty rectangular area at the bottom of the window, likely intended for a plot or data output.

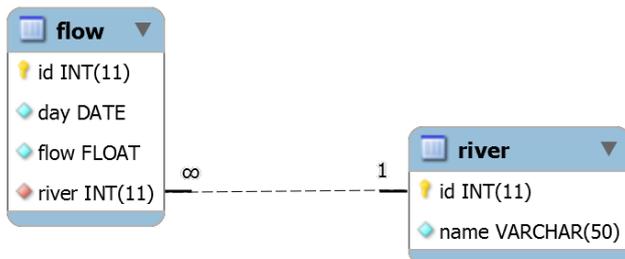
Nella realizzazione del codice, si lavori a partire dalle classi (Bean e DAO, FXML) ed il database contenuti nel progetto scaricabile. È ovviamente permesso aggiungere o modificare classi e metodi.

Tutti i possibili errori di immissione o validazione dati devono essere gestiti, non sono ammesse eccezioni generate dal programma.

Prova d'esame del 10/07/2015 – Turno B

Si consideri il data-set “**Rivers**”, ottenuto combinando un sottoinsieme delle informazioni presenti nella *Time Series Data Library*³ (TSDL). Tale data-set contiene la portata dei flussi di acqua di alcuni fiumi, misurati quotidianamente per diversi anni. Notare che i diversi fiumi sono stati misurati in intervalli di tempo indipendenti tra loro. Si noti anche che le portate dei fiumi sono molto diverse tra loro, di vari ordini di grandezza.

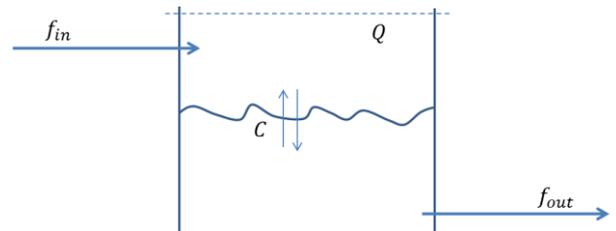
La struttura del database è la seguente:



La tabella ‘river’ contiene i fiumi considerati nel data set, ciascuno rappresentato da un identificatore numerico ‘id’ e da una stringa descrittiva ‘name’.

La tabella ‘flow’ contiene le misurazioni effettuate, nelle date specificate dal campo ‘day’ e per il fiume identificato dal campo (chiave esterna) ‘river’. La colonna ‘flow’ riporta il flusso misurato nel giorno specificato, espresso in *metri cubi per secondo*. La chiave primaria ‘id’ è un semplice identificatore univoco della misurazione.

Si intende costruire un’applicazione JavaFX che permetta di interrogare tale base dati, e supportare un progettista che ha l’incarico di costruire la diga per la realizzazione di un bacino idrico che si inserisce nel flusso del fiume (v. schema a fianco).



Si intende costruire un’applicazione JavaFX che permetta di interrogare tale base dati, e supportare un progettista che ha l’incarico di costruire la diga per la realizzazione di un bacino idrico che si inserisce nel flusso del fiume.

L’applicazione dovrà svolgere le seguenti funzioni:

1. Permettere all’utente di selezionare, da una tendina, uno specifico fiume. Appena il fiume è selezionato, occorre riempire i campi dell’interfaccia che riportano, rispettivamente: il valore minimo del flusso misurato per tale fiume, il valore medio del flusso misurato per tale fiume (detto f_{med}), il valore massimo del flusso misurato per tale fiume, il numero totale di giorni per cui esistono delle misurazioni.
2. Permettere all’utente di simulare l’effetto di un bacino idrico della capienza totale Q (espressa in metri cubi). La simulazione dovrà aggiornare, giorno per giorno, la quantità di acqua presente nel bacino (detta C), in funzione dei dati sul flusso in ingresso (detto f_{in}) presente nel data set. Il calcolo deve tenere conto delle seguenti regole:
 - a. Il valore di Q è impostato⁴ a $Q = f_{med} \cdot 30 \text{ giorni} \div 2$, ossia a metà dell’acqua che viene immessa in un mese “medio”.
 - b. L’occupazione iniziale del bacino è $C = Q/2$.
 - c. Il bacino dovrà garantire un flusso in uscita (detto f_{out}) pari ad almeno f_{out_min} , ogni giorno dell’anno.
 - d. L’utente può specificare il valore di f_{out_min} inserendo un “fattore di scala” $k > 0$, dal quale si calcherà $f_{out_min} = k \cdot f_{med}$.
 - e. Quando $f_{in} > f_{out}$, il livello C nel bacino salirà di una quantità legata alla differenza tra i due flussi

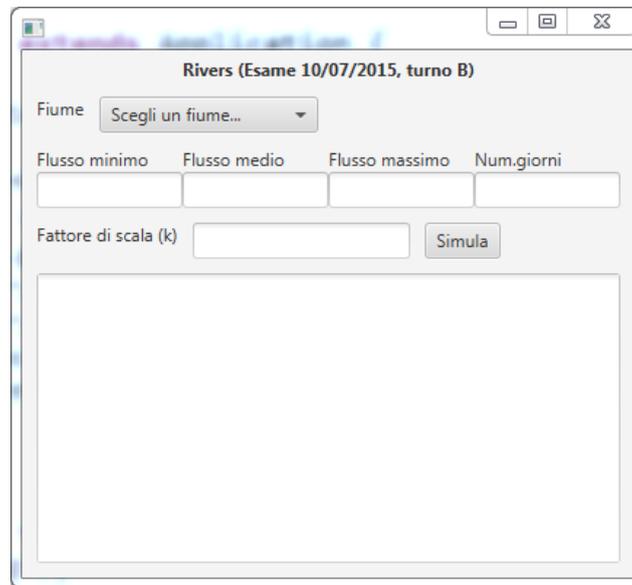
³ <https://datamarket.com/data/list/?q=interval:day%20provider:tSDL>, create da Rob Hyndman, Professore di Statistica alla Monash University, Australia

⁴ Attenzione alle unità di misura, in quanto f_{med} è espresso in m^3 al secondo e non al giorno

- f. Non è possibile avere $C > Q$. In caso di flusso in ingresso eccessivo, questo deve essere scaricato in uscita (evento detto “tracimazione”).
- g. In ogni giorno dell’anno, c’è una probabilità pari al 3% di dover azzerare il flusso in uscita, per poter compiere dei lavori nell’alveo del fiume.

Il simulatore dovrà determinare, in funzione di k :

- il numero di giorni in cui avviene una tracimazione;
- l’occupazione media C_{med} del bacino nel corso della simulazione.



Nella realizzazione del codice, si lavori a partire dalle classi (Bean e DAO, FXML) ed il database contenuti nel progetto scaricabile. È ovviamente permesso aggiungere o modificare classi e metodi.

Tutti i possibili errori di immissione o validazione dati devono essere gestiti, non sono ammesse eccezioni generate dal programma.