Basi di Dati, Assignment 3, 2020/21 **Scadenza: Dec 21, 11:56pm CET**

Database

```
Sailor (id:int, name:char(50), address:char(50), age:int, level:float)
Boat (bid:char(25), bname:char(50), size:char(30), captain:int)
```

- `captain` è una chiave esterna (FK) a `Sailor`
- Nessun attributo può essere `NULL`.

Task

Creare un programma Java, o se preferite, Python 3, che si connetta al database descritto sopra, ed esegua le seguenti operazioni, nel seguente ordine:

1. Fa il drop delle due tabelle dalla base di dati se sono già presenti.
2. Crea le due tabelle come descritto sopra.
3. Genera 1 milione di tuple (casuali[1]), in modo tale che ogni tupla abbia un valore diverso per l'attributo level, e le inserisce nella tabella `Sailor`. Assicurarsi inoltre che l'ultima tupla inserita, e solo quella, abbia come valore dell'attributo level, il valore 185.
4. Genera 1 ulteriore milione di tuple (casuali) e le inserisce nella tabella `Boat`.
5. Ottiene dal database tutti gli `id` del milione di tuple della tabella `Sailor` e li stampa su stderr.
6. Tutte le tuple con valore di level pari a 185 vengono modificate, cambiando il valore di level a 200 (la vostra query dovrà funzionare anche se la base di dati contiene più di una tupla con valore di level pari a 185).
7. Seleziona l'id e l'address di tutte le tuple della tabella Sailor che hanno valore di level pari a 200, e li stampa su stderr.
8. Crea un indice B+tree sull'attributo level.
9. Ottiene dal database tutti gli `id` del milione di tuple della tabella `Sailor` e li stampa su stderr.
10. Tutte le tuple con valore di level pari a 200 vengono modificate, cambiando il valore di level a 210 (la vostra query dovrà funzionare anche se la base di dati contiene più di una tupla con valore di level pari a 200).
11. Seleziona l'id e l'address di tutte le tuple della tabella Sailor che hanno valore di level pari a 210, e li stampa su stderr.

Per ognuna delle operazioni descritte sopra, si dovrà riportare (stampando su stdout) il tempo richiesto per eseguire l'operazione. Per farlo, è possibile mantenere in una variabile l'istante precedente l'inizio dell'esecuzione (in nanosecondi), poi ottenere il tempo corrente di sistema successivo al completamento dell'esecuzione, e calcolare la differenza in nanosecondi. Questo sarà approssimativamente il tempo richiesto per eseguire l'operazione. L'output su stdout dovrà essere della seguente forma:

```
Step 1 needs 10 ns
Step 2 needs 27 ns
Step 3 needs 77 ns
…
```

Note & Consegna

- Dovrete consegnare UN solo file (Java o Python) chiamato A3_XXX.java (oppure .py) dove XXX è la vostra matricola.
- La consegna verrà effettuata tramite il seguente Google form https://forms.gle/UXju8cEiVSps6P1j9
- Potete assumere che JDBC sia già presente nel classpath (java) e che il modulo psycopg2 sia già installato (Python 3).

[1] Notare che 1,2,3,4,5, … non verrà considerata come sequenza random valida. Potete utilizzare un qualunque generatore di numeri pseudo-casuali tra quelli messi a disposizione dal vostro linguaggio. Se preferite, potete anche decidere di implementarne uno.