

# Esame del 20/07/2021

Si consideri il database **yelp.sql**, estratto dal popolare sito web <https://www.yelp.it/> e contenente informazioni su locali come ristoranti e bar (Business), sugli utenti di queste attività commerciali (Users), e sulle loro recensioni (Reviews). Esso contiene informazioni per l'intervallo temporale tra l'anno 2005 e 2013 compresi, ed è strutturato secondo il diagramma ER illustrato nella pagina seguente.

**N.B.** Il database non è incluso nel progetto base, ma si può scaricare dal link <https://github.com/TdP-datasets/Yelp/releases/download/1.0/yelp.SQL.zip>

Si intende costruire un'applicazione JavaFX che permetta di svolgere le seguenti funzioni:

## PUNTO 1

a. Permettere all'utente:

- di inserire un numero  $n$  di recensioni;
- di selezionare un anno  $a$  compreso tra il 2005 e il 2013.

b. Alla pressione del bottone "Crea Grafo", si crei un grafo semplice, non orientato e pesato, i cui nodi siano tutti gli utenti (tabella *Users*) che abbiano pubblicato almeno  $n$  recensioni, **indipendentemente dall'anno inserito**.

N.B.: per ogni utente, si considerino le recensioni effettivamente contenute nella tabella *Reviews*. Il campo "review\_count" della tabella *Users* **non va considerato**.

c. Due utenti *diversi* vanno collegati da un arco se e solo se, almeno una volta, essi abbiano pubblicato una recensione per uno stesso locale commerciale nell'anno  $a$  selezionato. Il peso dell'arco, sempre positivo, è dato dal numero di volte che questo accade, e rappresenta il **grado di similarità** tra i due utenti coinvolti.

d. Permettere all'utente di selezionare, dall'apposita tendina, un utente  $u$  tra quelli presenti nel grafo. Alla pressione del bottone "Utente Simile" stampare, se esiste, l'utente collegato più simile ad  $u$ , ovvero quello con **grado di similarità** maggiore. In caso ci sia più di un utente che abbia lo stesso grado di similarità con  $u$ , stamparli tutti.

The screenshot shows a JavaFX window titled "Esame 20/07/2021 (Yelp)". It contains several input fields and buttons. At the top, there's a label "# recensioni (n)" followed by a text input field and a "Crea Grafo" button. Below that is a label "Anno (a)" followed by a dropdown menu. Then a label "Utente (u)" followed by another dropdown menu and a "Utente Simile" button. Further down, there's a label "Intervistatori (x1)" followed by a text input field and a "Simula" button. At the bottom, there's a label "Utenti (x2)" followed by a text input field. The main area of the window is currently empty.

**N.B.:** la dimensione del database utilizzato è considerevole, e alcune combinazioni di parametri iniziali (numero di recensioni  $n$  e anno  $a$ ) potrebbero risultare computazionalmente dispendiosi. Si faccia riferimento ai valori riportati negli screenshot dell'ultima pagina di questo documento per testare il programma.

## PUNTO 2

Una agenzia di marketing vuole condurre delle interviste ai clienti dei ristoranti e bar presenti sulla piattaforma Yelp per tracciare un profilo degli utenti e progettare delle campagne pubblicitarie mirate. Si vuole creare un programma che permetta all'agenzia di simulare tale analisi, in modo tale da poter pianificare in anticipo il numero di dipendenti ("intervistatori") da coinvolgere e il tempo necessario per portare a termine il lavoro. In particolare, il programma dovrà svolgere le seguenti funzioni:

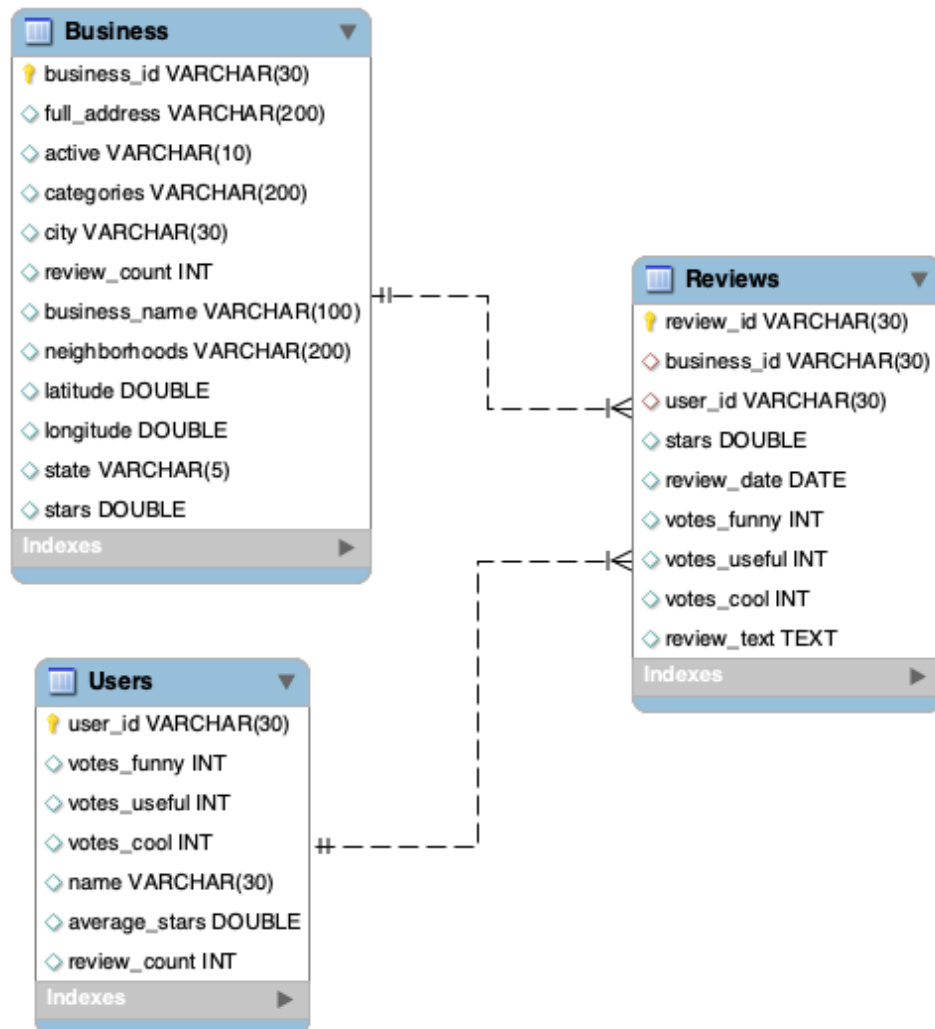
a. Permettere l'inserimento, negli appositi campi di testo, di un numero  $x1$  di intervistatori disponibili e un numero  $x2$  di utenti da intervistare. Il numero di utenti da intervistare deve essere minore o uguale del numero di utenti inclusi nel grafo costruito al punto 1, mentre il numero di intervistatori è sempre (molto) minore del numero di utenti da intervistare.

- b. Ogni intervistatore intervista al massimo 1 utente al giorno, e ogni utente viene intervistato al massimo una volta durante l'analisi di mercato. Inizialmente, l'agenzia assegna ad ognuno degli  $x1$  intervistatori un utente diverso da intervistare il primo giorno, scegliendo a caso tra gli utenti inclusi nel grafo costruito al punto 1.
- c. Alla fine di ogni giorno, ogni intervistatore relaziona l'agenzia sul suo lavoro in vista della giornata successiva:
  - I. Nel 60% dei casi, l'intervistatore ha portato a termine la propria intervista, e l'agenzia gli assegna un nuovo utente da intervistare per il giorno successivo. Il nuovo utente assegnato è sempre, se possibile, l'utente non ancora intervistato più simile a quello appena intervistato (ovvero quello con **grado di similarità** maggiore, si veda il punto 1d). Nel caso in cui ci siano più di un utente con lo stesso grado di similarità, l'utente da intervistare è scelto a caso da questo insieme. Nel caso invece in cui non ci siano utenti simili, ad esempio perché l'utente in questione non è collegato ad altri vertici o perché tutti gli utenti collegati sono già stati intervistati, il nuovo utente da intervistare è scelto a caso tra tutti gli utenti ancora da intervistare.
  - II. Nel 20% dei casi, invece, l'intervistatore è riuscito a portare a termine la propria intervista, ma chiede un giorno di ferie per il giorno successivo. In questo caso, l'agenzia concede il giorno di ferie e non assegna nessun nuovo utente all'intervistatore per il giorno seguente. Al giorno seguente, si procederà alla scelta dell'intervistato secondo i criteri del punto I. (che quindi sarà scelta al 100%).
  - III. Nel rimanente 20% dei casi, infine, l'intervistatore non è riuscito a portare a termine la propria intervista. In questo caso, l'intervistatore proverà a intervistare di nuovo lo stesso utente il giorno successivo.
- d. La simulazione termina quando l'agenzia, grazie al lavoro dei suoi dipendenti, riesce ad intervistare esattamente  $x2$  utenti. Al termine della simulazione si stampi:
  - I. Il numero di utenti intervistati da ogni intervistatore.
  - II. Il numero di giorni che sono serviti a portare a termine l'analisi di mercato.

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 **Business**, **Users** e **Reviews** contengono rispettivamente informazioni su locali commerciali, utenti e loro recensioni. Una recensione è collegata al locale commerciale di riferimento (tabella **Business**) tramite la chiave esterna **business\_id**, e all'utente che l'ha creata (tabella **Users**) tramite la chiave esterna **user\_id**.



## ESEMPI DI RISULTATI PER CONTROLLARE LA PROPRIA SOLUZIONE:

Esame 20/07/2021 Yelp

# recensioni (n)

Anno (a)

Utente (u)

Intervistatori (x1)

Utenti (x2)

Grafo creato con 44 vertici e 58 archi

Esame 20/07/2021 Yelp

# recensioni (n)

Anno (a)

Utente (u)

Intervistatori (x1)

Utenti (x2)

Utenti più simili a Alexander (IPaYMDmJbAnv\_3pmZH\_inw):

Jake (Ovpa3S8xD96dLE5eDxcxJg)	GRADO: 2
Stevey (4_YfclHkCem1onJFktxJA)	GRADO: 2
Alicia (APUPfqt1R8QyHhHk2uAyA)	GRADO: 2
Kristin (f53FUDHRHLg7BQ89KgAtxQ)	GRADO: 2

Esame 20/07/2021 Yelp

# recensioni (n)

Anno (a)

Utente (u)

Intervistatori (x1)

Utenti (x2)

Grafo creato con 82 vertici e 494 archi

Esame 20/07/2021 Yelp

# recensioni (n)

Anno (a)

Utente (u)

Intervistatori (x1)

Utenti (x2)

Utenti più simili a Heather (wHg1YkCzdZq9WBJOTRgxHQ):

Michael (90a6z--_CUri84aCzZyPsg)	GRADO: 11
----------------------------------	-----------