

# Esame del 23/02/2022

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

- Permettere all'utente di selezionare, dall'apposita tendina, una città *c*. Per avere l'elenco delle città disponibili, si faccia riferimento alla colonna *city* nella tabella *Business*.
- Quando l'utente seleziona una città *c*, popolare la tendina con l'elenco di tutti i locali commerciali presenti in tale città, permettendo all'utente di selezionare un locale commerciale *b*. A tal proposito, si utilizzi il metodo *doRiempilocali* del controller già incluso nel progetto base, che viene invocato ogni qual volta l'utente seleziona una nuova città.
- Alla pressione del bottone "Crea Grafo", si crei un grafo **semplice, orientato e pesato**, i cui nodi siano tutte le recensioni (tabella *Reviews*) del locale commerciale *b*.
- Ogni recensione *r* va collegata (eventualmente) con tutte le sue recensioni **consecutive**, ovvero con le recensioni che hanno una data di pubblicazione (colonna *review\_date*, tabella *Reviews*) più recente<sup>1</sup>:
  - gli archi devono essere orientati dalla recensione *r* verso le recensioni più recenti;
  - il peso di ogni arco (sempre positivo) è dato dalla differenza, in termini di giorni<sup>2</sup>, tra la data di pubblicazione di *r* e la data di pubblicazione della recensione più recente; se tale differenza è 0, l'arco non deve essere inserito.
- Si trovi nel grafo la recensione per cui il numero di archi uscenti sia massimo. Si stampino l'identificativo della recensione (colonna *review\_id*, tabella *Reviews*) e il relativo numero di archi uscenti. Nel caso in cui ci sia più di una recensione che abbia lo stesso numero di archi uscenti, il programma deve stamparle tutte.

## PUNTO 2

Alla pressione del bottone "Trova Miglioramento", si sfrutti il grafo creato al punto 1 per trovare e stampare la più lunga sequenza di recensioni consecutive per cui il punteggio di ogni recensione (colonna *stars*, tabella *Reviews*) sia sempre **migliore o uguale** di quello della recensione precedente. Insieme alla sequenza di recensioni, si stampi anche il numero complessivo di giorni che intercorrono tra la prima e l'ultima recensione della sequenza, sfruttando le informazioni codificate negli archi del grafo.

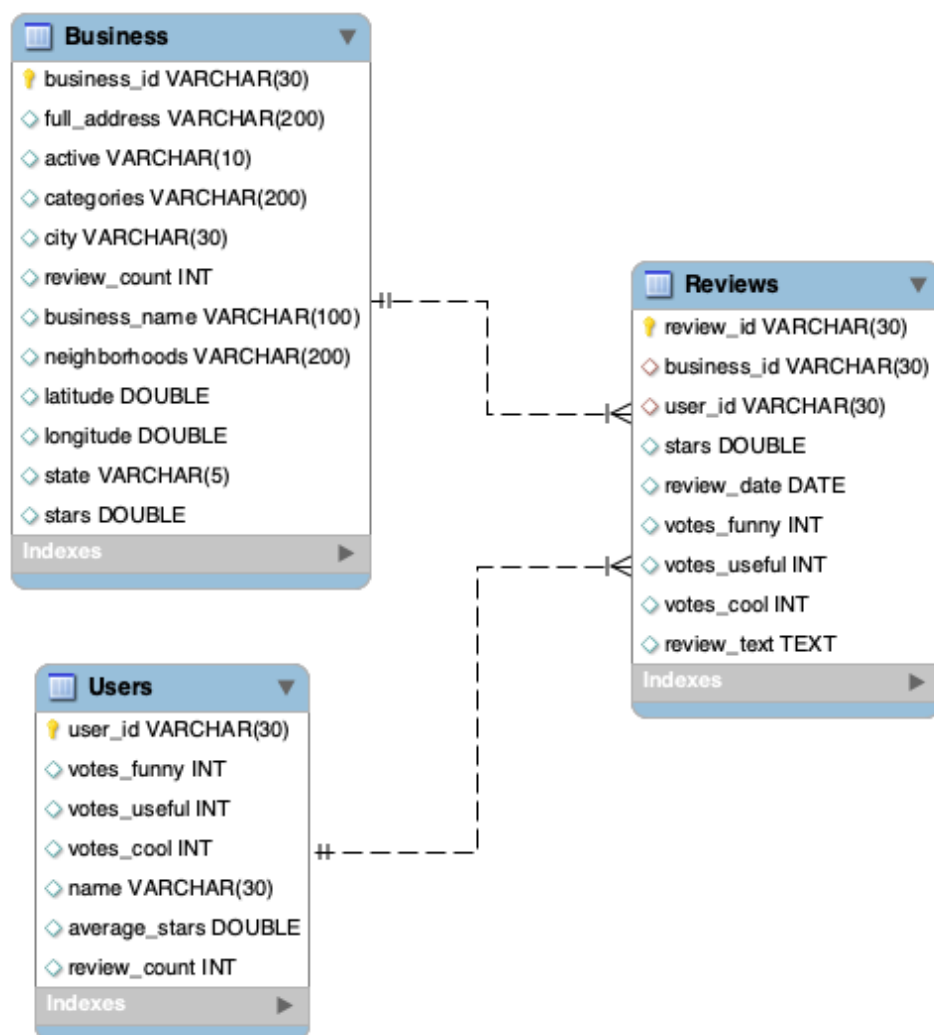
<sup>1</sup> Per confrontare due oggetti di tipo *LocalDate* si possono utilizzare i metodi *isBefore* e/o *isAfter* <https://docs.oracle.com/javase/8/docs/api/java/time/LocalDate.html>

<sup>2</sup> Per trovare la differenza in giorni tra due oggetti di tipo *LocalDate* si può utilizzare il metodo *ChronoUnit.DAYS.between(LocalDate date1, LocalDate date2)* <https://docs.oracle.com/javase/8/docs/api/java/time/temporal/ChronoUnit.html>

Nella realizzazione del codice, si lavora 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:**

The image displays three examples of the application's output for different city and locale combinations. Each example is shown in a window titled "Esame 23/02/2022 Yelp".

**Example 1 (Top Left):**

- Città (c): Avondale
- Locale (b): Guitar Center
- Grafo creato: 5 vertici, 10 archi
- T1NTDWH9wDCeQwKBdc11w # ARCHI USCENTI: 4

**Example 2 (Top Right):**

- Città (c): Scottsdale
- Locale (b): Zpizza
- Grafo creato: 34 vertici, 559 archi
- vraZSndBEYeTm9zaGuxtw # ARCHI USCENTI: 33

**Example 3 (Bottom):**

- Città (c): Phoenix
- Locale (b): 25 Degrees
- Grafo creato: 59 vertici, 1710 archi
- KT4mCvTZM64xiEUndgPoTQ # ARCHI USCENTI: 58