Σχεδίαση και Χρήση Βάσεων Δεδομένων

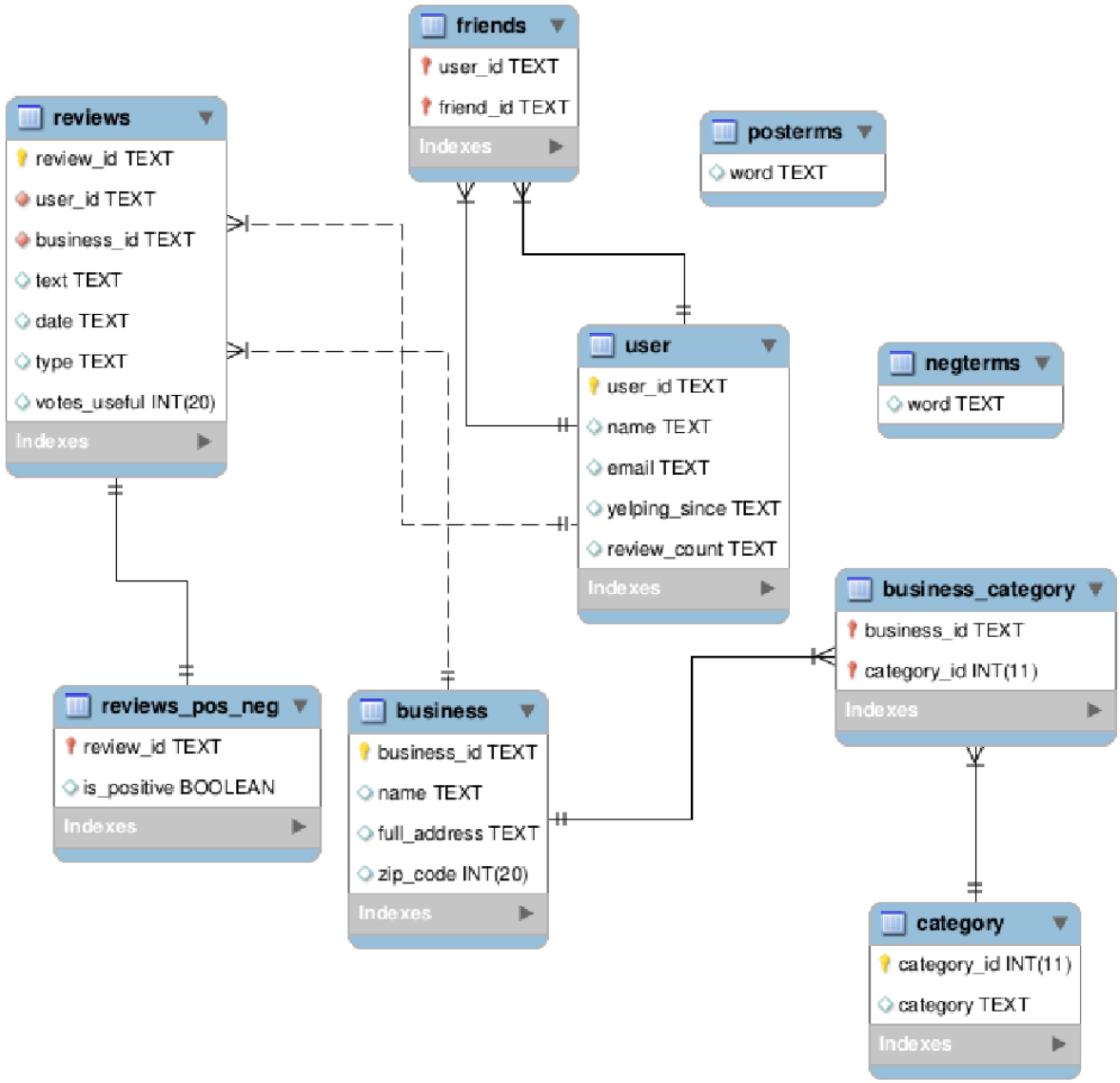
Εαρινό Εξάμηνο 2019-2020

2η Άσκηση

Παράδοση Άσκησης: 14 Ιουνίου 2020, 23:55

Δεδομένα

Σε αυτήν την εργασία, σας δίνεται μια βάση δεδομένων η οποία περιέχει τις αξιολογήσεις / κριτικές των επιχειρήσεων στον ιστοχώρο https://www.yelp.com. Το σχήμα της βάσης είναι το παρακάτω:



Μέρος 1ο - Ερωτήματα SQL

1. Βρείτε τους χρήστες με το όνομα 'Lisa' για τους οποίους ο αριθμός των αναφορών είναι μεγαλύτερος από 500 και ταξινομήστε τους με βάση την ημερομηνία χρήσης του yelp.
2. Ελέγξτε αν έγινε κάποια αναφορά από τον χρήστη 'Lisa' στην επιχείρηση 'Gab & Eat'.
3. Ελέγξτε για την επιχείρηση με κωδικό 'OmpbTu4deR3ByOo7btTTZw' αν υπάρχουν θετικές αναφορές.Το ερώτημα θα πρέπει να επιστρέφει ως απάντηση μια σχέση με μια πλειάδα και μια στήλη με τιμή “yes” ή “no”. Απαγορεύεται η χρήση Flow Control Operators (δηλαδή, if, case, κλπ.)
4. Βρείτε πόσες επιχειρήσεις μέσα στο έτος 2014 έχουν περισσότερες από δέκα θετικές ή αρνητικές αναφορές.
5. Βρείτε πόσες αναφορές έχει κάνει κάθε χρήστης στην κατηγορία 'Mobile Phones'.
6. Βρείτε τους χρήστες που έκαναν αναφορές για την επιχείρηση 'Midas', ταξινομώντας τα αποτελέσματα με βάση χρήσιμες ψήφους σε φθίνουσα σειρά.

Μέρος 2ο – Υλοποίηση διεπαφής με Python

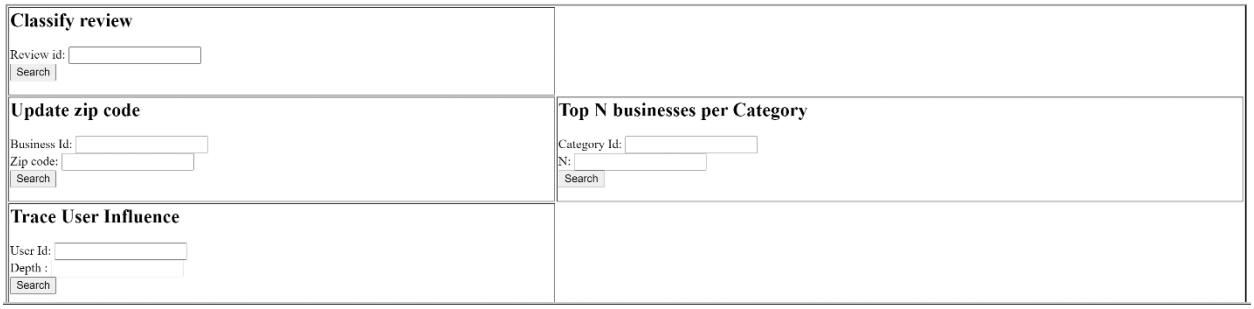
Σας δίνεται επίσης και μια εφαρμογή τριών επιπέδων. Αποτελείται από τη διεπαφή χρήστη, που είναι web--based, τη λογική της εφαρμογής, που είναι σε Python, και τη βάση δεδομένων, που είναι σε MySQL.

Οδηγίες:

Για να την τρέξετε, θα πρέπει να κάνετε τα εξής:

- να κάνετε unzip το application.zip
- να τρέξετε το website.py με την python
- να ανοίξετε κάποιον browser και να βάλετε τη διεύθυνση “http://localhost:8080”

Το παρακάτω είναι η αρχική σελίδα που πρέπει να δείτε στην εφαρμογή:



Αυτό που καλείστε να κάνετε είναι να αλλάξετε τη λογική της εφαρμογής η οποία βρίσκεται στο αρχείο **app.py** έτσι ώστε να εκτελεί τα παρακάτω ζητούμενα. Όλες οι παραπάνω συναρτήσεις επιστρέφουν μια **λίστα από πλειάδες (tuples)** όπου πάντα η πρώτη πλειάδα είναι η κεφαλίδα με τα ονόματα των πεδίων και οι υπόλοιπες είναι τα αποτελέσματα.

Για παράδειγμα: [(“Name”, “Id”,),(“Jim”,7,),(“Tom”,13,)]

Περιγραφή των συναρτήσεων:

1. **classify_review*:** Η συνάρτηση αυτή παίρνει ως όρισμα τον κωδικό μιας αξιολόγησης. Βρίσκει το πλήθος των θετικών ή αρνητικών όρων (**posterms**, **negterms**) που υπάρχουν μέσα στο κείμενο της αξιολόγησης και ταξινομεί μία αξιολόγηση ως θετική ή αρνητική. Για να το βρει αυτό, παίρνει το κείμενο της αξιολόγησης λέξη προς λέξη και βρίσκει τους θετικούς ή αρνητικούς όρους που υπάρχουν μέσα σε αυτό. Οι όροι αποτελούνται από μία ή δύο ή τρεις λέξεις.
   Το παραπάνω μπορεί να γίνει με ένα (ή περισσότερα) ερώτημα SQL χρησιμοποιώντας επιπλέον και μία συνάρτηση **extract_ngrams** που θα γράψετε σε python. Η συνάρτηση **extract_ngrams** θα είναι αυτής της μορφής:

   extract_ngrams (text, num) όπου text είναι ένα review και num ένας αριθμός 1,2,3 που αντιστοιχεί στον τύπο των n-grams που θέλετε να εξάγετε.
   Για παράδειγμα, για τη φράση 'A class is a blueprint for the object.' έχουμε τους παρακάτω όρους:
   1-gram: ['A', 'class', 'is', 'a', 'blueprint', 'for', 'the', 'object']
   2-gram: ['A class', 'class is', 'is a', 'a blueprint', 'blueprint for', 'for the', 'the object']
   3-gram: ['A class is', 'class is a', 'is a blueprint', 'a blueprint for', 'blueprint for the', 'for the object']

   Η **classify_review** επιστρέφει:

   - Το όνομα της επιχείρησης.
   - Το θετικό ή αρνητικό σχόλιο.

   Hint: για να βγουν σωστά αποτελέσματα έχει σημασία ο αριθμός των λέξεων στη φράση που έχει βρεθεί. Αν η θετική φράση περιέχει 3 λέξεις τότε ο μετρητής θετικών φράσεων πρέπει να αυξηθεί κατά 3. Επίσης, όταν μια φράση έχει ήδη μετρηθεί, π.χ.,“good food”, τότε δεν πρέπει να μετριέται ξεχωριστά ο όρος “good” που είναι substring της αφού έχει ήδη μετρηθεί σαν λέξη της πλήρους φράσης.

   Στη συνάρτηση **classify_review** απαγορεύεται να χρησιμοποιήσετε τη σχέση reviews_pos_neg. Η σχέση αυτή περιέχει για κάθε κριτική το αν είναι θετική ή αρνητική (0,1) και μπορείτε να τη χρησιμοποιήσετε μόνο για να ελέγξετε τα αποτελέσματα σας αλλά και σε επόμενα ερωτήματα που είναι απαραίτητη.

2. **update_zip_code**: Η συνάρτηση αυτή παίρνει από το χρήστη ως ορίσματα την ταυτότητα μιας επιχείρησης και ενημερώνει το πεδίο zip_code της διεύθυνσης αυτής με την τιμή που δίνει ο χρήστης. Σε περίπτωση επιτυχίας, επιστρέφει ‘ok’. Αντίθετα, αν δεν υπάρχει τέτοια ταυτότητα επιστρέφει ‘error’.

3. **selectTopNbusinesses**: Η συνάρτηση αυτή δέχεται ως όρισμα τον κωδικό μιας κατηγορίας και έναν ακέραιο N. Βρίσκει τις N επιχειρήσεις ανά κατηγορία με βάση το πλήθος των θετικών αξιολογήσεων. Επιστρέφει τα εξής:
   - Τον κωδικό της επιχείρησης.
   - Το πλήθος των θετικών αξιολογήσεων για την κάθε επιχείρηση.

   Το παραπάνω μπορεί να γίνει με ένα (ή περισσότερα) ερώτημα SQL.

4. **traceUserInfluence**: Η συνάρτηση αυτή δέχεται ως όρισμα τον κωδικό ενός χρήστη (user_id). Στη συνέχεια, υπολογίζει το μεταβατικό εγκλεισμό (Transitiveclosure) της επιρροής του χρήστη αυτού σε άλλους χρήστες ως προς την επιλογή επιχειρήσεων και επιστρέφει τους κωδικούς των χρηστών που επηρεάζει. Το παραπάνω μπορεί να γίνει με ένα (ή περισσότερα) ερώτημα SQL.

   Hint: πώς επηρεάζεται ένας χρήστης από έναν άλλον; Ένας χρήστης **a** επηρεάζει ένα χρήστη **b** όταν είναι φίλοι και έχουν αξιολογήσει την ίδια επιχείρηση με την αξιολόγηση του **a** να προηγείται χρονικά του **b**. Η επιρροή αυτή δημιουργεί ένα γράφο όπου οι κόμβοι του είναι χρήστες και οι κατευθυνόμενες ακμές του αναπαριστούν επιρροές (υπάρχει κατευθυνόμενη ακμή ανάμεσα σε δύο κόμβους **a** και **b** αν ο **a** επηρέασε τον **b**). Αυτό που θέλουμε είναι να υπολογίσουμε το μεταβατικό εγκλεισμό αυτού του γράφου (http://en.wikipedia.org/wiki/Transitive_closure). Το αποτέλεσμα περιέχει τους χρήστες που επηρεάζει ο δεδομένος χρήστης. Αν, π.χ. ο **a** έχει επηρεάσει τον **b** και ο **b** έχει επηρεάσει τον **c**, τότε και ο **a** έχει επηρεάσει τον **c** (έμμεσα). Δηλαδή, αν a→b και b→c, τότε στο αποτέλεσμα πρέπει να εμφανίζονται τα εξής:
   - (b,)
   - (c,) // Λόγω μεταβατικού εγκλεισμού

   Σημείωση: υποθέτουμε ότι το σύνολο των δεδομένων είναι μεγάλο και δεν μπορεί να χωρέσει ολόκληρο στη μνήμη. Δεδομένου αυτού, θα πρέπει να προσπελάσετε το υποσύνολο των δεδομένων που χρειάζεται να εμφανιστούν στο αποτέλεσμα (θα πρέπει να εκτελέσετε παραπάνω από ένα ερώτημα στην βάση). Επίσης σας δίνεται σαν όρισμα το βάθος στο οποίο θα ψάξετε. Για παράδειγμα, αν το βάθος είναι 1 τότε θα ψάξετε μόνο τους φίλους του συγκεκριμένου χρήστη ενώ αν το βάθος είναι 2 θα ψάξετε μέχρι και τους φίλους των φίλων του χρήστη.

Γενικά ζητήματα που πρέπει να προσέξετε

- Είναι σημαντικό να μελετήσετε/τρέξετε τα παραδείγματα των εργαστηρίων πριν ξεκινήσετε την υλοποίηση της εργασίας.
- Για το **2ο μέρος** της εργασίας, στο αρχείο app.py έχετε έναν σκελετό για την εργασία και θα χρειαστεί να τροποποιήσετε τις συναρτήσεις που δίνονται ώστε να υλοποιηθούν τα ζητούμενα. Αντιθέτως τα υπόλοιπα αρχεία στον φάκελο application.zip δεν χρειάζεται να τροποποιηθούν.
- Το **2ο μέρος** της εργασίας πρέπει να εμφανίζει τα αποτελέσματα στον browser, σε μία διεύθυνση όπως πχ η “http://localhost:8080”.

Παραδοτέα

Η άσκηση θα γίνει από τις ίδιες ομάδες των 2 ή 3 ατόμων με τις οποίες εκπονήσατε την Άσκηση 1. Το ίδιο μέλος της ομάδας θα υποβάλλει στην η-Τάξη:

- Την εργασία σας σε ένα zip αρχείο. Το zip αρχείο θα έχει όνομα AM1_AM2[_AM3].zip όπου AM1 ο αριθμός μητρώου του 1ου μέλους της ομάδας που αναλαμβάνει να υποβάλλει την εργασία, AM2 ο αριθμός μητρώου του 2ου μέλους και, εφόσον υπάρχει, AM3 ο αριθμός μητρώου του 3ου μέλους.

Τι θα περιέχει το zip αρχείο:

1. Ένα readme.txt αρχείο όπου θα αναφέρετε τα μέλη της ομάδας (ονοματεπώνυμο – Α.Μ.).
2. Για τα ερωτήματα sql στο **1ο μέρος,** ένα αρχείο sql.txt στο οποίο θα βάλετε τα SQL ερωτήματα το ένα κάτω από το άλλο προσθέτοντας μπροστά από κάθε ερώτημα το νούμερο της ερώτησης που απαντά.
3. Για το **2ο μέρος,** ΜΟΝΟ το αρχείο app.py.

Η προθεσμία παράδοσης της εργασίας είναι **14 Ιουνίου 2020**, **23:55**, μόνο μέσω του e-class, στην περιοχή Εργασίες > ΣΧΒΔ 2019-2020: ΆΣΚΗΣΗ 2.

Καλή επιτυχία!