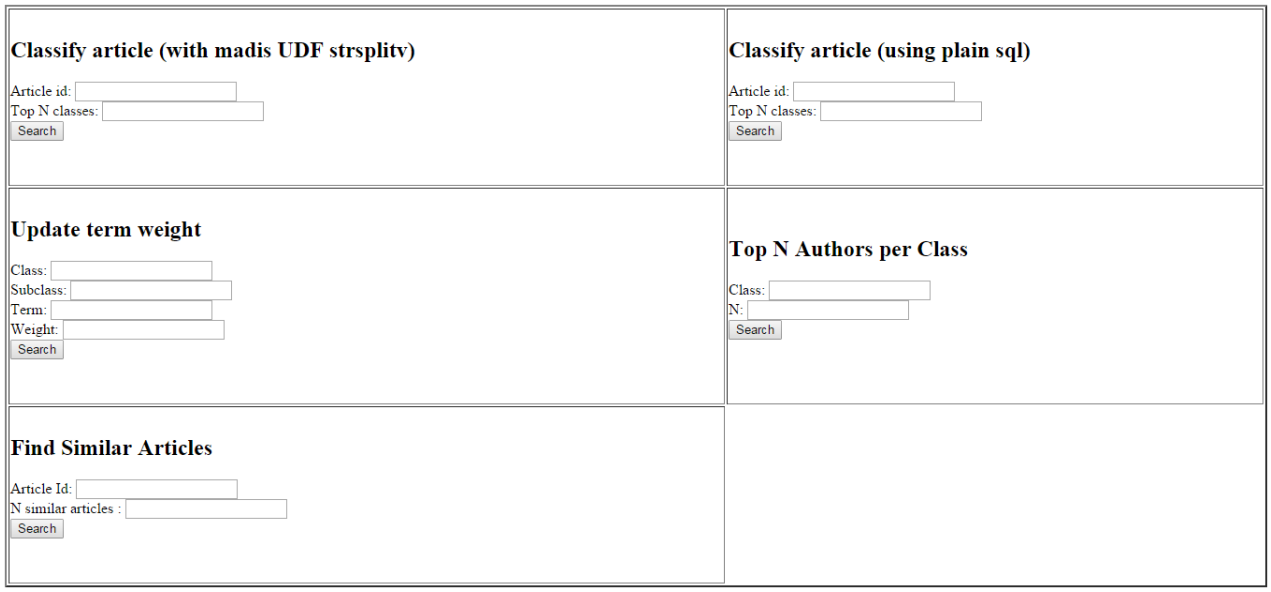
Σχεδίαση και Χρήση Βάσεων Δεδομένων
Εαρινό Εξάμηνο 2015
3η Άσκηση
Χτίζοντας μια εφαρμογή με Python και SQLite (+madIS)
Παράδοση: 25 Μαΐου 2015

Οδηγίες

Σας δίνεται μια εφαρμογή τριών επιπέδων. Αποτελείται από τη διεπαφή χρήστη, που είναι web-based, τη λογική της εφαρμογής, που είναι σε python, και το madIS, που είναι ένα σύστημα διαχείρισης βάσεων δεδομένων χτισμένο πάνω στην SQLite). Για να την τρέξετε, θα πρέπει να κάνετε τα εξής:

- Προτεινόμενη πλατφόρμα υλοποίησης : Linux- Ubuntu
- Να εγκαταστήσετε το madIS ακολουθώντας τις οδηγίες στο http://doc.madis.googlecode.com/hg/install.html. Τεκμηρίωση για το madIS μπορείτε να βρείτε στον σύνδεσμο http://doc.madis.googlecode.com/hg/index.html.
- να αποσυμπιέσετε με unzip το application.zip
- να αλλάξετε το αρχείο settings.py και να βάλετε το τοπικό μονοπάτι που βρίσκεται το madIS στον υπολογιστή σας.
- Να τρέξετε το website.py με την python όπου θα χρησιμοποιήσετε τη γραμμή εντολών
- να ανοίξετε κάποιον browser και να βάλετε τη διεύθυνση "http://localhost:9090"

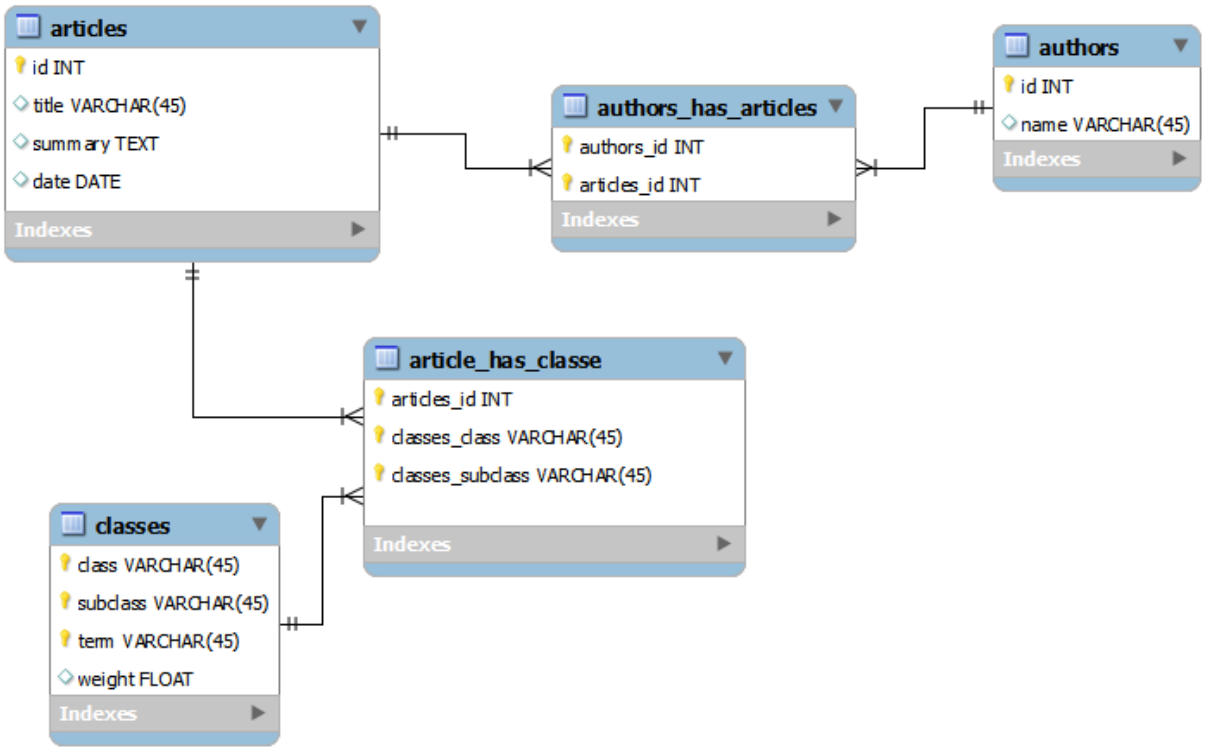
Το παρακάτω είναι η αρχική σελίδα που πρέπει να δείτε:



Αυτό που καλείστε να κάνετε είναι να αλλάξετε τη λογική της εφαρμογής η οποία βρίσκεται στο app.py (συγκεκριμένα, τις συναρτήσεις classify, classify_plain_sql, updateweight, selectTopNauthors και findSimilarArticles) έτσι ώστε να εκτελεί τα παρακάτω ζητούμενα. Όλες οι συναρτήσεις επιστρέφουν μια λίστα από πλειάδες 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,)]

Το σχήμα της βάσης SQLite που θα χρησιμοποιήσετε είναι το παρακάτω και η βάση σας αποτελείται από το αρχείο articles.db:



Σε αυτή την εργασία, σας δίνεται μια βάση δεδομένων η οποία περιέχει (μετα)πληροφορίες για επιστημονικά άρθρα. Θα χρειαστεί να χρησιμοποιήσετε μέσα στα SQL ερωτήματα συναρτήσεις ορισμένες από το χρήστη (User Defined Functions) που υπάρχουν υλοποιημένες στο madIS. Στο πλαίσιο της άσκησης καλείστε αρχικά να υλοποιήσετε κάποιες συναρτήσεις σε Python, οι οποίες επικοινωνούν με τη βάση δεδομένων και πραγματοποιούν κάποιες συγκεκριμένες λειτουργίες. Επιπλέον, προκειμένου να επιλύσετε το ερώτημα 5 θα χρειαστεί ακόμη να υλοποιήσετε ένα UDF εσωτερικά στο madIS.

Περιγραφή των συναρτήσεων

1. **classify:** Η συνάρτηση αυτή παίρνει ως όρισμα τον κωδικό ενός άρθρου και έναν ακέραιο N. Βρίσκει τις N κλάσεις που είναι οι πιο σχετικές με το περιεχόμενο του άρθρου και τις επιστρέφει ταξινομημένες. Για να το βρει αυτό, παίρνει την περίληψη του άρθρου λέξη προς λέξη και βρίσκει τις κλάσεις/υποκλάσεις με το μεγαλύτερο άθροισμα βαρών που προέρχεται από όρους οι οποίοι εμφανίζονται στην περίληψη του άρθρου. Το παραπάνω μπορεί να γίνει με ένα ερώτημα SQL χρησιμοποιώντας σε αυτό και τον τελεστή strsplitv του madIS. Οι απαραίτητες πληροφορίες για τον τελεστή strsplitv είναι διαθέσιμες εδώ:

   http://doc.madis.googlecode.com/hg/row.html?highlight=strsplitv#functions.row.formating.strsplitv

   Επιστρέφει:
   - Τον τίτλο του άρθρου.
   - Το όνομα της κλάσσης.
   - Το όνομα της υποκλάσσης
   - Το άθροισμα βαρών

2. **classify_plain_sql:** Η συνάρτηση αυτή κάνει ό,τι και η παραπάνω χωρίς την χρήση τελεστών του madIS, χρησιμοποιώντας δηλαδή απλά ερωτήματα SQL και ό,τι υλοποίηση χρειάζεται στην πλευρά της python.

   Στις συναρτήσεις (1,2) απαγορεύεται να χρησιμοποιήσετε την σχέση article_has_class. Η σχέση αυτή περιέχει για κάθε άρθρο την πιο σχετική κλάση και μπορεί να την χρησιμοποιήσετε μόνο για να ελέγξετε τα αποτελέσματα σας αλλά και σε επόμενα ερωτήματα που είναι απαραίτητη.

3. **updateweight:** Η συνάρτηση αυτή παίρνει από το χρήστη ως ορίσματα τα ονόματα κλάσης, υποκλάσης και όρου και ενημερώνει το πεδίο weight του όρου αυτού με το μέσο όρο της τρέχουσας τιμής του στο πεδίο αυτό και της τιμής που δίνει ο χρήστης. Σε περίπτωση επιτυχίας, επιστρέφει ‘ok’. Αντίθετα, αν δεν υπάρχει τέτοιος όρος / κλάση ή υποκλάση, επιστρέφει ‘error’.

4. **selectTopNauthors:** Η συνάρτηση αυτή δέχεται ως όρισμα μια κλάση και έναν ακέραιο N. Βρίσκει τους N συγγραφείς που έχουν συγγράψει τα περισσότερα άρθρα της κλάσης αυτής. Επιστρέφει τα εξής:
   - Τον κωδικο του συγγραφέα.
   - Τον αριθμό των άρθρων της κλάσης αυτής που έχει συγγράψει.

5. **FindSimilarArticles:** Η συνάρτηση αυτή δέχεται ως όρισμα τον κωδικό ενός άρθρου και έναν ακέραιο N και επιστρέφει τα N πιο όμοια άρθρα με βάση τις περιλήψεις τους. Η ομοιότητα των περιλήψεων υπολογίζεται με βάση την ομοιότητα Jaccard (Jaccard Similarity) για την οποία υπάρχει στο madIS τελεστής υπολογισμού της.

   Η ομοιότητα Jaccard μπορεί να μην δώσει λογικά αποτελέσματα όταν στο κείμενο υπάρχουν πολλές κοινές λέξεις. Για αυτόν το λόγο, θα πρέπει να υλοποιηθεί στο madIS ο συναθροιστικός τελεστής findCommonTerms, ο οποίος από όλες τις περιλήψεις, βρίσκει το 10% των πιο συχνά χρησιμοποιούμενων λέξεων και επιστρέφει μια σχέση με μια στήλη, στην οποία κάθε πλειάδα περιέχει μια από αυτές τις λέξεις. Οι λέξεις αυτές θα πρέπει να μην συμμετέχουν στον υπολογισμό της ομοιότητας Jaccard.

   Η υλοποίηση του συναθροιστικού τελεστή findCommonTerms θα γίνει στο αρχείο findCommonTerms.py που περιέχεται στα συνοδευτικά αρχεία και στη συνέχεια θα μεταφέρετε το αρχείο στον κατάλογο madis/src/functions/aggregate/ ώστε να μπορείτε να το χρησιμοποιήσετε.

   Οι υπόλοιποι τελεστές στο madIS που πρέπει να χρησιμοποιήσετε για να εφαρμόσετε την παραπάνω περιγραφή είναι οι ακόλουθοι:

   1. Τελεστής πλειάδας strsplitv: Παίρνει σαν είσοδο μια συμβολοσειρά και επιστρέφει μια σχέση με μια στήλη που οι πλειάδες της περιέχουν της λέξεις που εμφανίζονται στην συμβολοσειρά.
      http://doc.madis.googlecode.com/hg/row.html?highlight=strsplitv#functions.row.formating.strsplitv
   2. Συναθροιστικός τελεστής jgroup: Παίρνει σαν είσοδο μια σχέση και επιστρέφει όλες τις πλειάδες σε μια λίστα json.
      http://doc.madis.googlecode.com/hg/aggregate.html?highlight=jgroup#functions.aggregate.jpacks.jgroup
   3. Τελεστής πλειάδας jaccard: Παίρνει σαν είσοδο δύο λίστες json και επιστρέφει την ομοιότητα Jaccard μεταξύ τους.
      http://doc.madis.googlecode.com/hg/row.html?highlight=jaccard#functions.row.similarity.jaccard

Γενικά ζητήματα που πρέπει να προσέξετε

- Ρυθμίστε τον κειμενογράφο που χρησιμοποιείτε έτσι ώστε το tab να ισοδυναμεί με 4 κενά, αλλιώς θα έχετε προβλήματα με τη στοίχιση της Python.
- Σε ενημερώσεις της βάσης, για να αποθηκευτεί η αλλαγή σας, χρειάζεται μετά το ερώτημά σας να γράψετε την εντολή con.commit().
- Θα υπάρχει βαθμολογική μείωση σε περίπτωση που η βάση σας δεν είναι προστατευμένη από sql injection.
- Για να μπορεί να διορθωθεί και να βαθμολογηθεί ο κώδικάς σας θα πρέπει τουλάχιστον να μην έχει συντακτικά λάθη. Για αυτό πριν την υποβολή καλό είναι να κάνετε έναν τελευταίο έλεγχο!

Η άσκηση αυτή είναι ομαδική 2-3 ατόμων!

Καλή σας επιτυχία!