

# Σχεδίαση και Χρήση Βάσεων Δεδομένων - Εαρινό Εξάμηνο 2019

## 3<sup>η</sup> Άσκηση

**Παράδοση Άσκησης: Δευτέρα 03 Ιουνίου 2019, Ώρα: 23:59**

Σε αυτή την εργασία, σας δίνεται μια βάση δεδομένων η οποία περιέχει (μετα)πληροφορίες για επιστημονικά άρθρα. Στο πλαίσιο της άσκησης καλείστε να υλοποιήσετε κάποιες συναρτήσεις σε Python, οι οποίες επικοινωνούν με τη βάση δεδομένων και πραγματοποιούν κάποιες συγκεκριμένες λειτουργίες. Συγκεκριμένα, σας δίνεται μια εφαρμογή τριών επιπέδων που 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 (το σχήμα της βάσης και το dataset είναι τα ίδια με αυτά που χρησιμοποιήσατε στην Άσκηση 2).

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

- να μελετήσετε τις διαφάνειες του φροντιστηρίου για την άσκηση αυτή (εκεί θα βρείτε οδηγίες εγκατάστασης για την python, για το web framework που θα χρησιμοποιήσετε που ονομάζεται bottle, καθώς και βασικά παραδείγματα εξοικείωσης με μια web-based εφαρμογή)
- να κάνετε unzip το application.zip.
- να αλλάξετε το αρχείο settings.py και να βάλετε τα στοιχεία σας για να μπορείτε να συνδεθείτε στη βάση MySQL
- Στη συνέχεια να τρέξετε το website.py με την python
- να ανοίξετε κάποιον browser και να βάλετε τη διεύθυνση "<http://localhost:8080>"

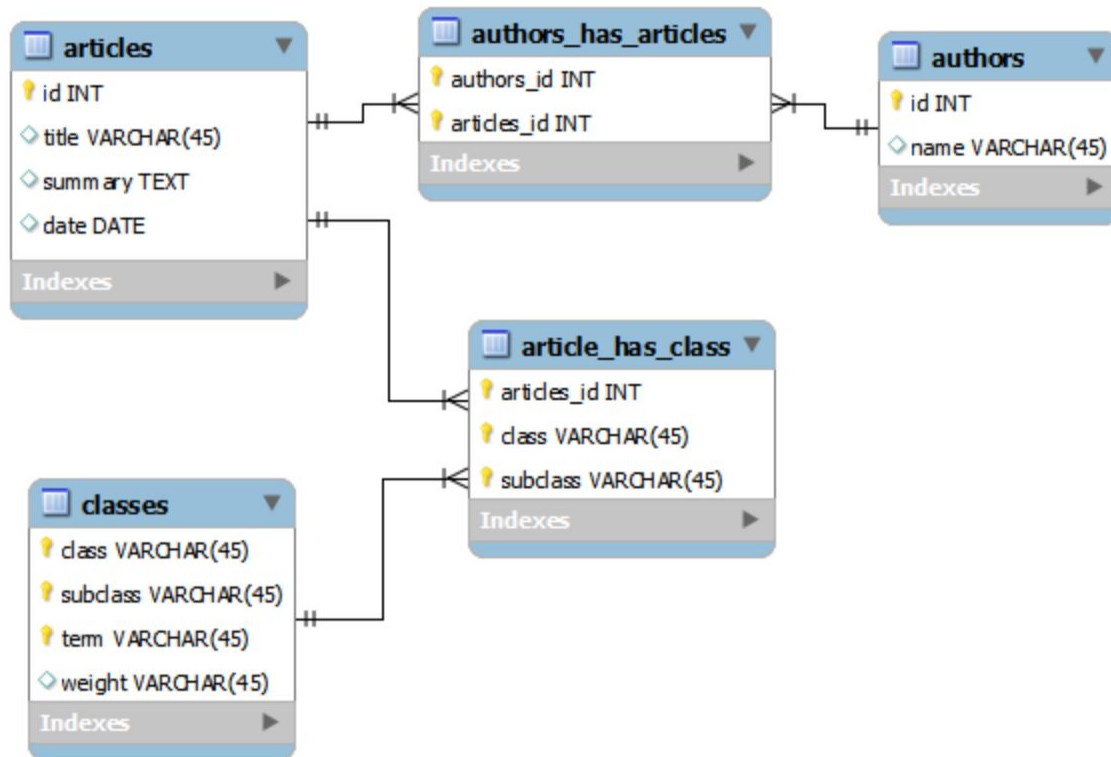
Το παρακάτω είναι η αρχική σελίδα που πρέπει να δείτε:

<b>Classify</b> Top N classes: <input type="text"/> <input type="button" value="Search"/>	<b>FindSimilarArticles</b> Article Id: <input type="text"/> N similar articles: <input type="text"/> <input type="button" value="Search"/>
<b>Update weight</b> Class: <input type="text"/> Subclass: <input type="text"/> Weight: <input type="text"/> <input type="button" value="Search"/>	<b>Top N Classes</b> From: <input type="text"/> To: <input type="text"/> N: <input type="text"/> <input type="button" value="Search"/>
<b>Count Articles</b> Class: <input type="text"/> Subclass: <input type="text"/> <input type="button" value="Search"/>	

Αυτό που καλείστε να κάνετε είναι να αλλάξετε τη λογική της εφαρμογής η οποία βρίσκεται στο **app.py** (συγκεκριμένα, τις συναρτήσεις classify, updateweight, κτλ) 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,)]

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



## Περιγραφή των συναρτήσεων:

**classify:** Η συνάρτηση αυτή παίρνει ως όρισμα έναν ακέραιο N. Βρίσκει αρχικά όλα τα άρθρα που δεν έχουν κατηγορία (κλάση και υποκλάση) και στη συνέχεια για κάθε άρθρο: βρίσκει τις N κατηγορίες που είναι οι πιο σχετικές με το περιεχόμενο του άρθρου και τις επιστρέφει ταξινομημένες. Μια κατηγορία αποτελείται από την κλάση και την υποκλάση. 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.

Επιστρέφει:

- Τον τίτλο του άρθρου.
- Το όνομα της κλάσης.
- Το όνομα της υποκλάσης
- Το άθροισμα βαρών

**updateweight:** Η συνάρτηση αυτή παίρνει από το χρήστη ως ορίσματα τα ονόματα κλάσης, υποκλάσης και ένα αριθμό 'W' και ενημερώνει το πεδίο weight των όρων με βάρος μεγαλύτερο από W, μειωνοντας τον κατά το μισό της διαφοράς τους (πχ. αν W=1

και  $w1 = 1,20$ , τότε το νέο  $w1$  θα είναι  $1,10$ ). Σε περίπτωση επιτυχίας, επιστρέφει 'ok'. Αντίθετα, αν δεν υπάρχει τέτοια κλάση ή υποκλάση, επιστρέφει 'error'.

**selectTopNClasses:** Η συνάρτηση αυτή δέχεται ως όρισμα δύο ημερομηνίες (from & to) και έναν ακέραιο N. Βρίσκει τις N κατηγορίες (κλάση και υποκλάση) που είχαν τα περισσότερα άρθρα στο συγκεκριμένο χρονικό διάστημα. Επιστρέφει τα εξής:

- Το όνομα της κλάσης
- Το όνομα της υποκλάσης
- Το σύνολο των άρθρων (COUNT)

**countArticles:** Η συνάρτηση αυτή δέχεται ως ορίσματα την κλάση και την υποκλάση και επιστρέφει τον σύνολο των άρθρων για την κατηγορία (κλάση και υποκλάση) . Επιστρέφει το εξής:

- Το σύνολο των άρθρων (COUNT)

**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 θα σας δώσουμε πληροφορίες στο φροντιστήριο. Για να υπολογίσετε την ομοιότητα Jaccard θα χρειαστεί να γράψετε μία δική σας συνάρτηση σε python (προσοχή μην χρησιμοποιήσετε έτοιμη βιβλιοθήκη).

### Γενικά ζητήματα που πρέπει να προσέξετε

- Είναι σημαντικό να μελετήσετε τα παραδείγματα του φροντιστηρίου πριν ξεκινήσετε την υλοποίηση 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.
- Θα παραδώσετε μόνο το αρχείο **app.py**, αφού το βάλετε σε ένα αρχείο της μορφής sdixxx\_app.zip (τα υπόλοιπα αρχεία που περιέχονται στο φάκελο application.zip, δεν χρειάζεται να συμπεριληφθούν στο παραδοτέο της εργασίας). Η εργασία πρέπει να εμφανίζει τα αποτελέσματα στον browser, σε μία διεύθυνση όπως π.χ. η "<http://localhost:8080>".

Καλή Επιτυχία!