

# Σχεδίαση και Χρήση Βάσεων Δεδομένων

Εαρινό Εξάμηνο 2021

## 3<sup>η</sup> Άσκηση

Παράδοση Άσκησης: 6 Ιουνίου 2021, Ώρα: 23:59

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 είναι το ίδιο με αυτό που χρησιμοποιήσατε στην Άσκηση 2, ενώ τα δεδομένα είναι διαφορετικά και θα τα βρείτε στην ηλεκτρονική τάξη στον φάκελο της εργασίας).

Για να την τρέξετε, θα πρέπει να κάνετε τα εξής κατά σειρά βήματα:

1. 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)
2. να κάνετε unzip το application.zip.
3. να αλλάξετε το αρχείο settings.py και να βάλετε τα στοιχεία σας για να μπορείτε να συνδεθείτε στη βάση MySQL
4. να τρέξετε το website.py με την python
5. να ανοίξετε κάποιον browser και να βάλετε τη διεύθυνση "<http://localhost:8080>"

Το παρακάτω είναι η αρχική σελίδα που πρέπει να δείτε:

<b>Most common symptoms</b> Vax name: <input type="text"/> <input type="button" value="Search"/>	<b>Patient report</b> Patient: <input type="text"/> <input type="button" value="Search"/>
<b>Build a new block</b> blockfloor: <input type="text"/> <input type="button" value="Search"/>	<b>Find nurse</b> Floor: <input type="text"/> Number of patients: <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, να συμπληρώσετε τις υπάρχουσες συναρτήσεις) έτσι ώστε να εκτελεί τα παρακάτω ζητούμενα. Όλες οι παραπάνω συναρτήσεις πρέπει να 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. **findnurse:** Για κάθε νοσοκόμα/ο που έχει κάνει βάρδια σε όλες τις πτέρυγες του ορόφου X και έχει εξυπηρετήσει στα ραντεβού τουλάχιστον Ψ ασθενών, να βρεθεί ο κωδικός, το όνομά της/του και το πλήθος των μοναδικών ασθενών στον οποίων τον εμβολιασμό έχει παραστεί. Η συνάρτηση αυτή παίρνει ως ορίσματα τον όροφο X και τον αριθμό των ασθενών Ψ και επιστρέφει:
  1. Το όνομα της νοσοκόμας ή του νοσοκόμου
  2. Τον κωδικό της νοσοκόμας ή του νοσοκόμου
  3. Το πλήθος των μοναδικών ασθενών στον οποίων τον εμβολιασμό έχει παραστεί
2. **patientreport:** Η συνάρτηση αυτή παίρνει ως όρισμα το όνομα ενός ασθενούς και εξάγει μία αναφορά η οποία περιέχει το ιστορικό του. Για το ιστορικό, χρειάζονται τα παρακάτω:
  1. Το όνομα του γιατρού που πραγματοποιεί τη θεραπευτική αγωγή
  2. Το όνομα της νοσοκόμας ή του νοσοκόμου που παρευρίσκεται στη θεραπεία
  3. Το όνομα της θεραπευτικής αγωγής στην οποία υποβλήθηκε
  4. Το κόστος της θεραπευτικής αγωγής
  5. Η ημερομηνία που πήρε εξιτήριο
  6. Το δωμάτιο όπου έγινε η εισαγωγή, καθώς και ο όροφος και η πτέρυγα που ανήκε το δωμάτιο
3. **mostcommonsymptoms:** Η συνάρτηση αυτή παίρνει ως όρισμα το όνομα ενός εμβολίου **vax\_name**. Εντοπίζει τα πιο συχνά συμπτώματα με βάση τη συχνότητα εμφάνισης τους στη στήλη **symptoms**. Για να το βρει αυτό, παίρνει το κείμενο της περιγραφής των συμπτωμάτων λέξη προς λέξη, αφού έχουν αφαιρεθεί τα σημεία στίξης και τα stopwords<sup>1</sup> κι έχουν μετατραπεί όλα τα κεφαλαία σε πεζά γράμματα, και βρίσκει τους "όρους" που υπάρχουν μέσα σε αυτό. Οι όροι αποτελούνται από μία ή δύο ή τρεις λέξεις. Στη συνέχεια κατατάσσει τους όρους με βάση τη συχνότητα εμφάνισης τους κατά φθίνουσα σειρά (ο πιο συχνός όρος να εμφανίζεται πρώτος).

Το παραπάνω μπορεί να γίνει με ένα (ή περισσότερα) ερώτημα SQL, χρησιμοποιώντας επιπλέον και μία συνάρτηση **create\_ngrams** που θα γράψετε σε python (δεν επιτρέπεται να χρησιμοποιήσετε έτοιμες βιβλιοθήκες που εξάγουν τα ngrams). Η συνάρτηση **create\_ngrams** θα είναι της εξής μορφής:

**create\_ngrams** (text, num), όπου text είναι ένα symptom και num ένας αριθμός 1,2,3, που αντιστοιχεί στον τύπο των n-grams που θέλετε να εξάγετε.

Για παράδειγμα, για τη φράση “A few hours after the shot my left arm was VERY sore” έχουμε τους παρακάτω όρους (στο παρακάτω παράδειγμα δεν έχουν αφαιρεθεί τα stopwords):

1-gram: ['a', 'few', 'hours', 'after', 'the', 'shot', 'my', 'left', 'arm', 'was', 'very', 'sore']

2-gram: ['a few', 'few hours', 'hours after', 'after the', 'the shot', 'shot my', 'my left', 'left arm', 'arm was', 'was very', 'very sore']

3-gram: ['a few hours', 'few hours after', 'hours after the', 'after the shot', 'the shot my', 'shot my left', 'my left arm', 'left arm was', 'arm was very', 'was very sore']

Η **mostcommonsymbols** επιστρέφει:

1. Το όνομα του εμβολίου
  2. Τη λίστα με τους πιο συχνούς όρους
4. **buildnewblock**: Η συνάρτηση αυτή παίρνει από το χρήστη ως όρισμα έναν κωδικό για τον όροφο **blockfloor** και δημιουργεί μία νέα πτέρυγα (**blockcode**) με 1 έως 5 δωμάτια (**RoomNumber**).

Στη βάση δεδομένων σας, κάθε όροφος (blockfloor) μπορεί να έχει μέχρι 9 πτέρυγες (blockcode) και κάθε πτέρυγα μπορεί να έχει μέχρι 5 δωμάτια. Ο τετραψήφιος κωδικός ενός δωματίου σχηματίζεται από το συνδυασμό των blockfloor, blockcode, και έναν από τους αριθμούς (00,01,02,03,04) (πχ 6501, όπου 6 είναι το blockfloor, 5 είναι το blockcode και 01 ο αριθμός του δωματίου).

Η συνάρτηση αυτή πρέπει

- να ελέγχει αν επιτρέπεται στον όροφο **blockfloor** που δίνεται ως όρισμα από το χρήστη να προστεθεί μία νέα πτέρυγα, και
- αν ναι, να γίνονται **INSERT** στον πίνακα **block** η νέα πτέρυγα και στον πίνακα **room** ο αριθμός από N νέα δωμάτια της πτέρυγας (όπου N θα είναι ένας τυχαίος αριθμός από το 1 μέχρι το 5), διατηρώντας την κωδικοποίηση που αναφέραμε παραπάνω (για κάθε νέο δωμάτιο δηλαδή πρέπει να δείτε τι επιπλέον στοιχεία πρέπει να εισαχθούν εκτός από το **RoomNumber**).

Σε περίπτωση επιτυχίας, επιστρέφει 'ok'. Αντίθετα, αν δεν επιτρέπεται η προσθήκη νέας πτέρυγας, επιστρέφει 'error'.

### 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 zip αρχείο με όνομα `AM1_AM2[_AM3].zip` όπου AM1 ο αριθμός μητρώου του 1ου μέλους της ομάδας που αναλαμβάνει να υποβάλει την εργασία, AM2 ο αριθμός μητρώου του 2ου μέλους και, εφόσον υπάρχει, AM3 ο αριθμός μητρώου του 3ου μέλους. Θα κρατήσετε τις ίδιες ομάδες που σχηματίσατε για την Άσκηση 1. 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"](http://localhost:8080).

Καλή Επιτυχία!

---

<sup>i</sup> stopwords: εννοούμε λέξεις που χρησιμοποιούνται συχνά αλλά δεν προσφέρουν σημαντική πληροφορία (για παράδειγμα λέξεις όπως `this`, `the`, `is`, `at`, κτλ.). Θα δημιουργήσετε εσείς μια λίστα με τέτοιες λέξεις την οποία και θα χρησιμοποιήσετε για να καθαρίσετε τις προτάσεις από τη στήλη `symptoms` πριν τον υπολογισμό των ngrams