

3η Άσκηση

Χτίζοντας μια εφαρμογή με Python και MySQL

Παράδοση: 24 Μαρτίου, 15:59

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 τη βάση δεδομένων, που είναι σε MySQL (το σχήμα της βάσης και το dataset είναι τα ίδια με αυτά που χρησιμοποιήσατε στην δεύτερη άσκηση). Για να την τρέξετε, θα πρέπει να κάνετε τα εξής:

- να κατεβάσετε την python έκδοση 2.7
- 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
- να τρέξετε το website.py με την python
 - σε Windows: με 2 κλικ πάνω στο αρχείο website.py ή αν θέλετε μπορείτε να δοκιμάσετε και από τη γραμμή εντολών
 - σε Linux: 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:

Update Rank Rank1: <input type="text"/> Rank2: <input type="text"/> Movie title: <input type="text"/> <input type="button" value="Update"/>	Colleagues of Colleagues Actor Id 1: <input type="text"/> Actor Id 2: <input type="text"/> <input type="button" value="Search"/>
Actor Pairs Actor Id: <input type="text"/> <input type="button" value="Search"/>	Top N Actors N: <input type="text"/> <input type="button" value="Search"/>
Trace Actor Influence Actor Id: <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 updateRank, colleaguesOfColleagues, actorPairs, selectTopNactors και traceActorInfluence) 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 παραδειγμα: [{"Name", "Id",}, {"Jim", 7,}, {"Tom", 13,}]

Περιγραφή των συναρτήσεων

- updateRank:** Η συνάρτηση αυτή παίρνει από το χρήστη ως όρισμα δυο αριθμητικές τιμές (double από 0 μέχρι και 10) και το όνομα μίας ταινίας και ενημερώνει το πεδίο rank της ταινίας αυτής με το μέσο όρο της τρέχουσας τιμής της στο πεδίο αυτό (αν υπάρχει) και των δύο τιμών που έδωσε ο χρήστης, ή με τον μέσο όρο των δύο τιμών (αν η τιμή του rank απουσιάζει). Σε περίπτωση επιτυχίας, επιστρέφει 'ok'. Αντίθετα, αν δεν υπάρχει ταινία με αυτό το όνομα, ή αν υπάρχουν περισσότερες της μιας ταινίες με το ίδιο όνομα, ή αν οι αριθμητικές τιμές είναι εκτός ορίων, επιστρέφει 'error'.
- colleaguesOfColleagues:** Η συνάρτηση αυτή παίρνει ως όρισμα ένα ζευγάρι (a,b) από κωδικούς ηθοποιών (πεδίο id). Βρίσκει τις ταινίες που παίζουν μαζί δύο άλλοι ηθοποιοί (c,d) με την προϋπόθεση ότι ο c έχει παίξει σε ταινία μαζί με τον a και ο d έχει παίξει σε ταινία μαζί με τον b. Επιστρέφει τα εξής:
 - Τον τίτλο της κάθε ταινίας των (c,d).
 - Τον κωδικό του ηθοποιού c.
 - Τον κωδικό του ηθοποιού d.
 - Τον κωδικό του ηθοποιού a.
 - Τον κωδικό του ηθοποιού b.
- actorPairs:** Η συνάρτηση αυτή δέχεται ως όρισμα τον κωδικό ενός ηθοποιού. Βρίσκει τους ηθοποιούς ο καθένας από τους οποίους έχει παίξει **μόνο** σε διαφορετικά είδη ταινιών από αυτόν αλλά που μαζί μ' αυτόν έχει παίξει σε τουλάχιστον επτά διαφορετικά είδη. Επιστρέφει το εξής:
 - Τους κωδικούς των ηθοποιών που πληρούν την παραπάνω συνθήκη.
- selectTopNactors:** Η συνάρτηση αυτή δέχεται ως όρισμα έναν ακέραιο N. Για κάθε είδος ταινίας, βρίσκει τους N ηθοποιούς που έχουν παίξει στις περισσότερες ταινίες του είδους. Επιστρέφει τα εξής:
 - Το όνομα του είδους.
 - Τον κωδικό κάθε ηθοποιού.
 - Τον αριθμό των ταινιών του είδους στις οποίες έχει παίξει ο κάθε ηθοποιός.
- traceActorInfluence:** Η συνάρτηση αυτή δέχεται ως όρισμα τον κωδικό ενός ηθοποιού. Στην συνέχεια, υπολογίζει το μεταβατικό εγκλεισμό της επιρροής του ηθοποιού αυτού σε άλλους ηθοποιούς ως προς το είδος ταινιών και επιστρέφει τους κωδικούς των ηθοποιών που επηρεάζει.

Hint: Πώς επηρεάζεται ένας ηθοποιός από έναν άλλον;

Ένας ηθοποιός a επηρεάζει έναν ηθοποιό b όταν έχουν παίξει σε μια ταινία μαζί (ανεξάρτητα από το πόσοι άλλοι ηθοποιοί παίζουν) και ο b παίζει σε μια άλλη μεταγενέστερης χρονιάς ταινία ίδιου είδους. Η επιρροή αυτή δημιουργεί ένα γράφο όπου οι κόμβοι του είναι ηθοποιοί 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[1]. Το αποτέλεσμα περιέχει τους ηθοποιούς που επηρεάζει ο δεδομένος ηθοποιός.

Για παράδειγμα, αν ο a έχει επηρεάσει τον b και ο b έχει επηρεάσει τον c , τότε και ο a έχει επηρεάσει τον c (έμμεσα). Δηλαδή, αν $a \rightarrow b$ και $b \rightarrow c$, 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).

1. http://en.wikipedia.org/wiki/Transitive_closure

Γενικά ζητήματα που πρέπει να προσέξετε

- Ρυθμίστε το tab του editor σας να μπαίνει ως 4 κενά αλλιώς θα έχετε προβλήματα με το indentation 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.

Η άσκηση αυτή είναι ατομική!

Καλή σας επιτυχία!