

Αναφορά 2^{ης} Εργαστηριακής Άσκησης

Εργαλεία Ανάπτυξης Λογισμικού

&

Προγραμματισμός Συστημάτων

Ελευθεριάδης Δημήτριος 2015030067

Καρελιώτη Μαρία 2015030197

Ολόκληρο το πρόγραμμα υλοποιήθηκε από κοινού.

Πρόγραμμα computeSales.py

Στην δεύτερη εργαστηριακή άσκηση το ζητούμενο είναι η υλοποίηση ενός προγράμματος, που θα δέχεται ως είσοδο αρχεία αποδείξεων ελέγχοντας την ορθότητα και εκτυπώνοντας στατιστικά, σε python.

Επεξήγηση Επιλογής sqlite

Έπειτα από αρκετή έρευνα, καταλήξαμε στη χρήση της sqlite βιβλιοθήκης που παρέχει η python. Ο λόγος ήταν καθώς η sqlite έχει γραφτεί σε C, προσφέροντας έτσι πολύ πιο γρήγορες ταχύτητες σε αναζητήσεις, ελέγχους ακόμα και αποθήκευσης πληροφορίας ακόμη και σε τεράστια αρχεία.

Επίσης, η sqlite παρέχει τη **δυνατότητα προσωρινής αποθήκευσης όλης της βάσης στη μνήμη**, γεγονός που την καθιστά ακόμα γρηγορότερη σε σχέση με read/write στο δίσκο.

Για παράδειγμα ένα text αρχείο που περιέχει 13.217.667 γραμμές, το πρόγραμμά μας χρειάζεται μόλις 46 δευτερόλεπτα* για διάβασμα, ελέγχους ορθότητας και αποθήκευσης της πληροφορίας στη βάση. Η ανάκτηση και εμφάνιση δεδομένων από τη Database, χρειάζεται μόλις κλάσματα δευτερολέπτου.

Το πρόγραμμα καλό θα ήταν να τρέξει με την εντολή sudo, ώστε να παρατεθούν τα απαραίτητα δικαιώματα για τη βάση, ωστόσο δεν ανάλογα τα δικαιώματα του εκάστοτε χρήστη μπορεί να μην είναι απαραίτητο.

*Το αποτέλεσμα αυτό προέκυψε ως μέσος όρος επανειλημμένων tests του κώδικα σε laptop dell με επεξεργαστή Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz και 7860MB συνολικής μνήμης.

Λειτουργικότητα Προγράμματος

Αρχικά υλοποιήθηκε το menu με τις 4 επιλογές: 1 για το διάβασμά ενός αρχείου αποδείξεων , 2 για τα στατιστικά ενός προϊόντος , 3 για τα στατιστικά ενός ΑΦΜ και 4 για έξοδο, μέσα σε μια while loop.

Στην συνέχεια υλοποιήσαμε τις 3 συναρτήσεις για τις 3 επιλογές.

Για το διάβασμα του αρχείου αποδείξεων:

Ζητάμε το όνομα του αρχείου μέσα στην συνάρτηση και με την εντολή with ανοίγουμε το αρχείο για διάβασμά. Διαβάζοντας γραμμή-γραμμή ελέγχουμε κάθε φορά, την ορθότητά της και κατ' επέκτασιν της εκάστοτε απόδειξης. Όταν εντοπίσουμε την γραμμή που αναφέρεται στο ΑΦΜ, παίρνουμε το ΑΦΜ και το περνάμε από τον έλεγχο (θα αναλυθεί παρακάτω). Έτσι συνεχίζουμε για όλη την απόδειξη και βρίσκουμε για κάθε προϊόν το name, το quantity, το price, το total και τέλος το bill total price. Μετά ελέγχουμε εάν συναντάμε το διαχωριστικό των αποδείξεων και ξέρουμε εάν αλλάξαμε απόδειξη. Όλα αυτά τα στοιχεία τα περνάμε σε ένα temporary table και έπειτα αν όλα τα στοιχεία είναι σωστά τα περνάμε στον κυρίως πίνακα που αποθηκεύουμε όλες τις σωστές αποδείξεις.

Τους πίνακες αυτούς τους δημιουργούμε, καθώς η υλοποίηση των 2 άλλων επιλογών έγινε μέσω sqlite και δημιουργήσαμε μια τοπική βάση δεδομένων. Επομένως φτιάξαμε μια συνάρτηση για το connection με την τοπική βάση, δύο συνάρτησεις που δημιουργούμε τους πίνακες (μία για το temp table και μία για το bill table), δύο συναρτήσεις για να γίνει το insert στους πίνακες και μία συνάρτηση για να κάνουμε delete τα στοιχεία από τον temp table.

Για την εκτύπωση των στατιστικών ενός προϊόντος:

Όπως προαναφέρθηκε το ερώτημα αυτό υλοποιήθηκε με sqlite. Παρακάτω παρατίθεται ο κώδικας :

```
85 def second_question(conn, pr):
86     cur = conn.cursor()
87     cur.execute("SELECT AFM, SUM(total) FROM bills WHERE product_name=? GROUP BY AFM, product_name", (pr,))
88     pn=cur.fetchall()
89     for row in pn:
90         t=round(row[1],2)
91         afm=str(row[0])
92         print(afm,t)
```

Για την εκτύπωση των στατιστικών ενός ΑΦΜ:

Στην ίδια λογική με την προηγούμενη υλοποιήθηκε και αυτή η επιλογή. Παρατίθεται ο κώδικας :

```
94 def third_question(conn, afm):
95     cur = conn.cursor()
96     cur.execute("SELECT product_name, SUM(total) FROM bills WHERE AFM=? GROUP BY product_name", (afm,))
97     pn=cur.fetchall()
98     for row in pn:
99         t=round(row[1],2)
100         print(row[0],t)
101
102
```

Έλεγχοι

Για την υλοποίηση του προγράμματος χρειάστηκε να γίνουν αρκετοί έλεγχοι.

Ο πρώτος έλεγχος πρόκειται για το ΑΦΜ, δηλαδή εάν υπάρχει και αν είναι 10ψήφιο.

Επίσης έγινε έλεγχος για το αν η συνολική τιμή ενός προϊόντος ανάλογά με το quantity, αλλά και η τελική τιμή της απόδειξης είναι ορθές.

Ακόμα, ελέγχουμε αν ξεκινάει νέα απόδειξη, όπου και λαμβάνουμε τα απαραίτητα μέτρα, μηδενίζοντας κάποιες βοηθητικές μεταβλητές και αδειάζοντας τον temp table.

Αξίζει να σημειωθεί, ότι σε αν οποιοδήποτε σημείο της απόδειξης βρεθεί σφάλμα, τότε το πρόγραμμα παραλείπει όλες τις σειρές αυτής της απόδειξης, μέχρι να βρει την καινούρια.

Το πρόγραμμα μας δεν είναι case-sensitive χρησιμοποιώντας την συνάρτηση upper() όταν διαβάζουμε την είσοδο από το χρήστη για το προϊόν που θέλει να αναζητήσει.

Τα στατιστικά επίσης εκτυπώνονται με αύξουσα σειρά μέσω group by.

Έχουμε επίσης χρησιμοποιήσει exceptions όπου χρειάζεται. Ένα παράδειγμα είναι στην περίπτωση που σε γραμμή κάποιας απόδειξης λείπει η τιμή του προϊόντος, τότε κάνουμε catch το exception αυτό, και θέτουμε την απόδειξη ως λανθασμένη.

Συμπεράσματα

Συμπερασματικά, η python είναι μία από τις πιο εύχρηστες γλώσσες για file handling και σε συνδυασμό με τη built-in βιβλιοθήκη της sqlite τα αποτελέσματα των ερωτημάτων ήταν γρήγορα και συνεπή.