Database αλυσίδα γυμναστηρίου.

Ομαδική εργασία στα πλαίσια του μαθήματος Βάσεις Δεδομένων.

Ομάδα 32  
  
Κοτσώνης Ιωάννης Νεκτάριος

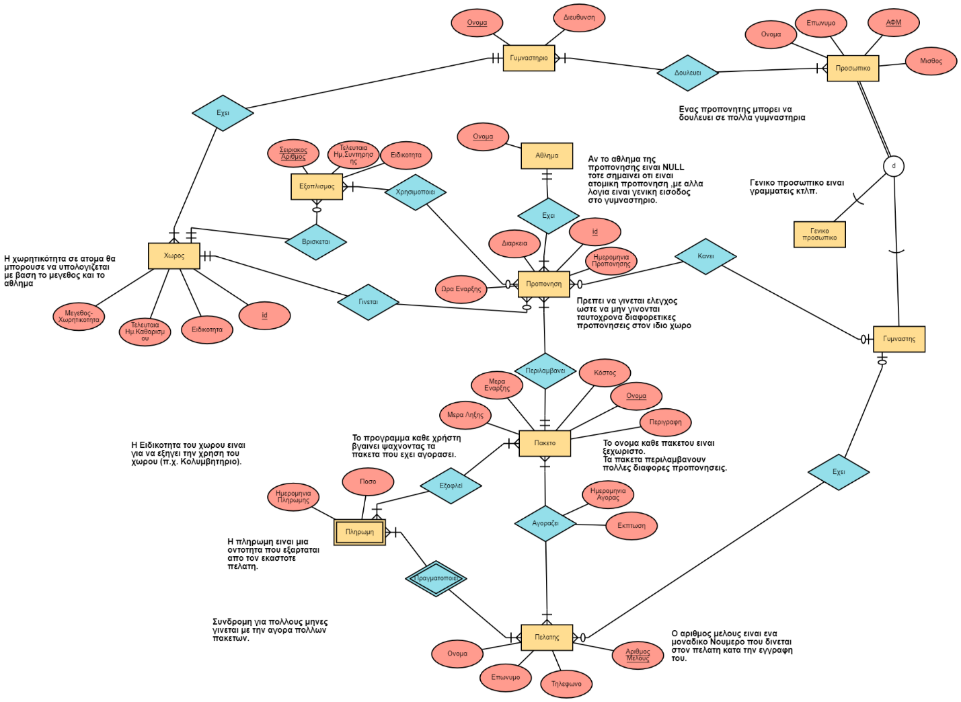
Α.Μ. :1066527  
Παπαδημητρίου Απόστολος

Α.Μ. 1066532

**Περίληψη**Στα πλαίσια του μαθήματος βάσεις δεδομένων μας δόθηκε η ομαδική εργασία με σκοπό την υλοποίηση μιας βάσης μιας αλυσίδας γυμναστηρίων. Σκοπός μας ήταν να φτιάξουμε μια βάση ώστε να συμπεριλάβουμε όσο το δυνατών περισσότερες οντότητες χρειάζεται για τον μικρόκοσμό μας χωρίς να υπερβούμε κάποιο παράλογο όριο. Επίσης για να είναι εύχρηστο και από έναν απλό χρήστη αναπτύχθηκε μια εφαρμογή σε python με την χρήση των βιβλιοθηκών tkinter και sqlite3 ώστε να δημιουργηθεί η βάση και ένα GUI για να το χρησιμοποιούν όλοι οι χρήστες.

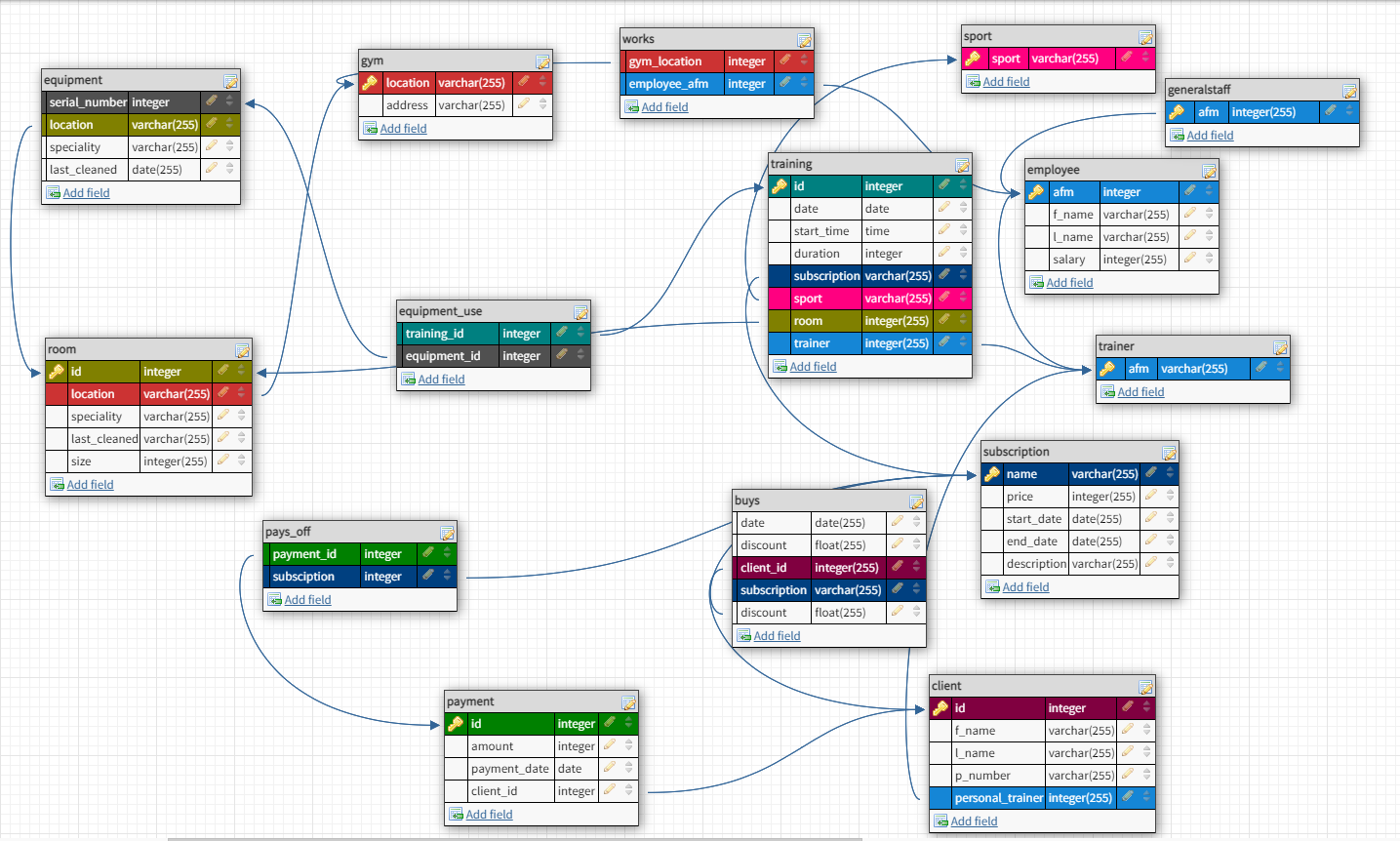
**Μεθοδολογία  
1. Παραδοχές**

Για την υλοποίηση του Project αρχικά κάναμε κάποιες παραδοχές του μικρόκοσμου όπου θα δουλεύαμε. Αρχικά η βάση μας αποθηκεύει τα απαραίτητα δεδομένα που έχουν να κάνουν με την άθληση του πελάτη όπως για παράδειγμα το πόσο ζυγίζει. Επίσης στην δικιά μας αλυσίδα αποφασίσαμε πως οι χώροι δεν θα μπορούν να νοικιαστούν από πελάτες και ότι δεν θα υπάρχουν συνεργασίες με αθλητικά σωμάτια.. Τέλος ο πελάτης με ενεργή συνδρομή μπορεί να επισκέπτεται όλα τα γυμναστήρια της αλυσίδας.

2.ERD  
Αφού κάναμε τις παραδοχές μας και αποφασίσαμε το τι θα αποθηκεύει η βάση μας ήμασταν σε θέση να δημιουργήσουμε το διάγραμμα οντοτήτων ERD. Το τελικό μας διάγραμμα αποτελείται από 11 οντότητες, οι πιο σημαντικές είναι ο πελάτης όπου σε αυτή θα αποθηκεύονται τα απαραίτητα στοιχεία κάθε πελάτη και η προπόνηση όπου από αυτή θα καθορίζετε εάν ο πελάτης θα έχει πρόσβαση στο γυμναστήριο.

ERD μικρόκοσμου

3. Relational Schema  
Με την δημιουργία του ERD ήμαστε σε θέση να φτιάξουμε το σχεσιακό μοντέλο βάση των σχέσεων που έχουν οι οντότητες μεταξύ τους. Για την ανάπτυξη του σχεσιακού μοντέλου κάναμε όλες τις οντότητες πίνακες και προσθέσαμε τα attributes τους, Μετά βάση τον σχέσεων που έχουν μεταξύ τους φτιάξαμε νέους πίνακες για όσες έχουν σχέση του τύπου N-M ενώ για τις σχέσεις 1-Ν και 1-1 απλώς προσθέσαμε τα ξένα κλειδιά της άλλης.  
Με την δημιουργία του σχήματος μπορούμε πλέον να αναπτύξουμε κώδικα σε SQL.

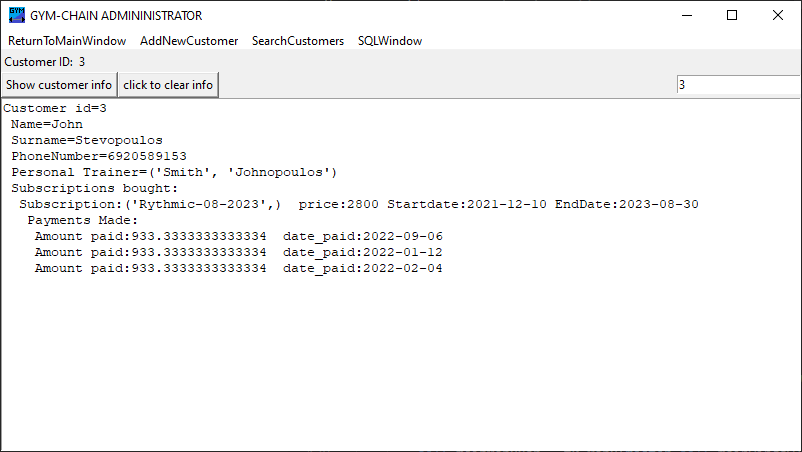


Relational Schema μικρόκοσμου.

4. SQL, Python και εφαρμογή αναζήτησης  
Φτάνοντας στο τελικό relational ήμαστε σε θέση να αναπτύξουμε την βάση μας μέσω python και με την βιβλιοθήκη sqlite3. Πρακτικά, η βιβλιοθήκη sqlite3 μας δίνει την δυνατότητα να γράψουμε SQL μέσα σε ένα python script. Αφού δημιουργήσαμε τα tables και τα γεμίσαμε με attributes, αρχίσαμε να γεμίζαμε με δεδομένα τους πίνακες. Αυτό έγινε από python script το οποίο διάβαζε κάποια .txt όπως το namesandSurnames ή το sportsandEquipment. Αφού είχαμε τα βασικά δεδομένα, με τυχαία επιλογή μέσω της random και σε συνδυασμό με τα queries της sqlite μπορέσαμε και γεμίσαμε με δεδομένα τους πίνακες. Όλα τα παραπάνω γίνονται στο αρχείο simplePopulator.py.   
  
Στη συνέχεια υλοποιήθηκε μια εφαρμογή με γραφική διεπαφή , η οποία θα απλοποιούσε την άντληση δεδομένων από τη βάση δεδομένων. Στο πλαίσιο της απλοποίησης δημιουργήθηκαν συνάρτησης και κλάσεις ‘Εργαλεία’ που θα επέτρεπαν την εκτέλεση πιο συνθέτων εντολών SQL με το πάτημα ενός κουμπιού. Επιπλέον δόθηκε σημασία στο να κάνουμε τις εντολές να αλλάζουν ευκολά παραμέτρους, ώστε να μπορεί και αργότερα η εφαρμογή να τροποποιηθεί για να εκτελεί και διαφορετικά queries με ελάχιστες αλλαγές στον κώδικα.

Στη συνέχεια παρουσιάζονται εν συντομία τα παράθυρα της εφαρμογής.

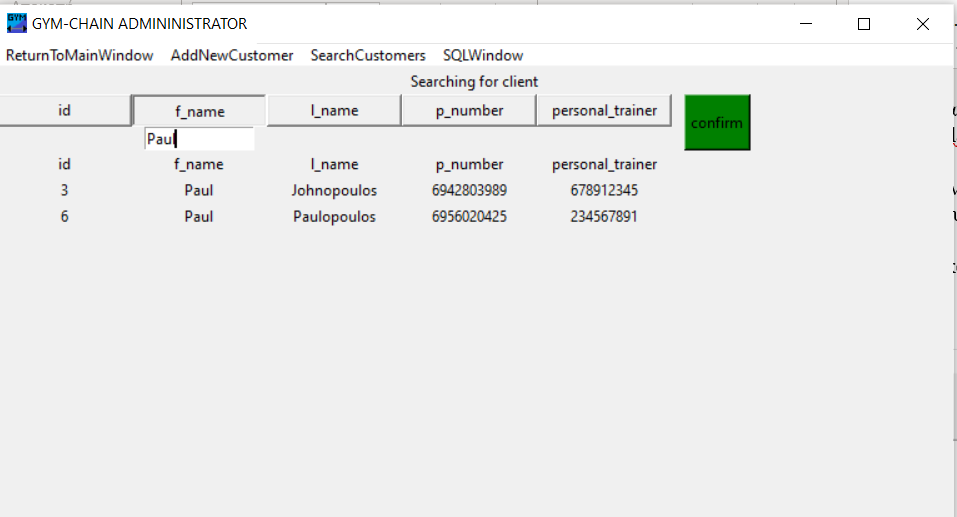
Η παρακάτω εικόνα δείχνει το κεντρικό παράθυρο της εφαρμογής. Σε αυτό ο υπάλληλος μπορεί να ψάξει κάποιον πελάτη βάση του ID του και θα του επιστρέψει τα στοιχεία του και τις συνδρομές του.

Στην εικόνα μπορούμε να διακρίνουμε τα βασικά στοιχεία του πελάτη. Επιπλέον παρατηρούμε ότι ο συγκεκριμένος πελάτης έχει αγοράσει ένα πακέτο και το έχει εξοφλήσει με 3 πληρωμές  
  


Κεντρικό Παράθυρο Εφαρμογής.

Στην επόμενη εικόνα φαίνεται το παράθυρο, στο οποίο ο υπάλληλος μπορεί να διαλέξει το φίλτρο αναζήτησης (όπως το όνομα, επίθετο κτλ. ), απλώς πατώντας στις παραμέτρους που επιθυμεί. Σημαντικό είναι να προσθέσουμε, πως το ίδιο παράθυρο μπορεί να χρησιμοποιηθεί για την παραμετρική αναζήτηση κι’ άλλων Οντοτήτων στην βάση δεδομένων. Το μόνο που πρέπει να αλλάξει είναι το όρισμα της συνάρτησης που επιλέγει σε ποια οντότητα θα γίνει αναζήτηση.

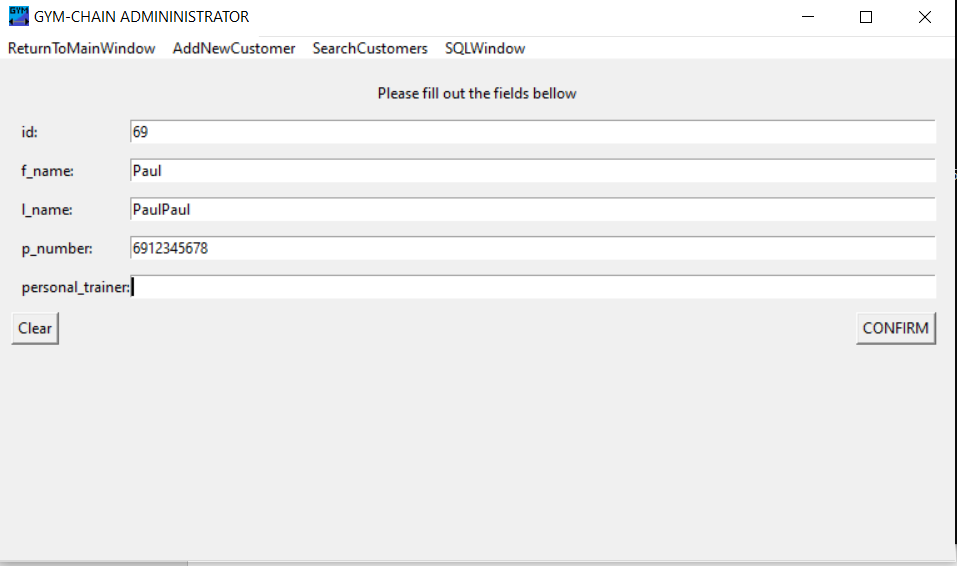
Για την συγκεκριμένη έκδοση επιτρέψαμε ,να μπορεί να γίνει αναζήτηση μόνο για τους πελάτες, μιας και αυτή είναι η πιο συχνή αναζήτηση που πραγματοποιείται στα γυμναστήρια. Στην εικόνα συγκεκριμένα βλέπουμε να έχει γίνει αναζήτηση για όλους τους πελάτες με όνομα “Paul”.



Παράθυρο Αναζήτησης.

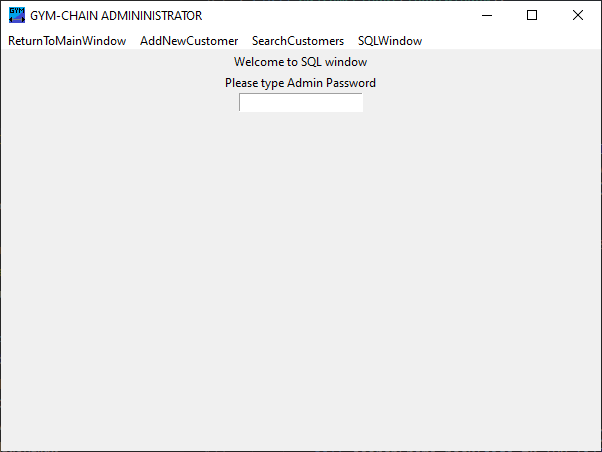
Στην επόμενη εικόνα φαίνεται το παράθυρο, στο οποίο ο υπάλληλος μπορεί να προσθέσει ένα νέο πελάτη συμπληρώνοντας τα βασικά του στοιχεία.

Παρόμοια με το παράθυρο αναζήτησης, το από κάτω παράθυρο μπορεί με μια απλή αλλαγή παραμέτρου να επιτρέψει την εισαγωγή κι άλλων οντοτήτων όπως προγράμματα προπονήσεων, προπονήσεις κλπ.

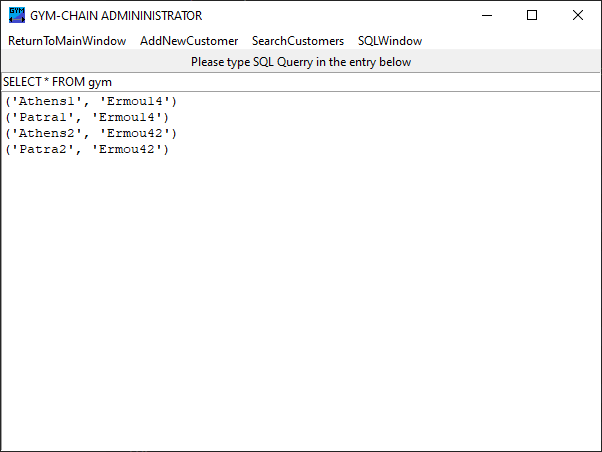


Παράθυρο εισαγωγής δεδομένων.

Τέλος στις δυο παρακάτω εικόνες είναι το παράθυρο SQL. Σε αυτό το παράθυρο οι υπεύθυνοι της αλυσίδας ή του γυμναστηρίου μπορούν να εκτελέσουν άμεσα μέσω της εφαρμογής εντολές SQL. Προκειμένου όμως να μην μπορεί οποιοσδήποτε χρήστης της εφαρμογής να εκτελέσει εντολές SQL (και πιθανά χαλάσει την βάση), το παράθυρο απαιτεί ένα κωδικό («BaseisDedomenwn2022») για να εμφανίσει το πεδίο για τις εντολές.



Παράθυρο SQL ,πριν πληκτρολογηθεί ο σωστός κωδικός



Παράθυρο SQL ,εφόσον πληκτρολογηθεί ο σωστός κωδικός

**Αξιολόγηση**Σκοπός του project εξαρχής ήταν η αναπαράσταση μιας βάσης οπού μπορεί να αποθηκεύσει τα βασικά στοιχεία ενός γυμναστηρίου και αυτά να αξιοποιηθούν από μια εφαρμογή. Στο τέλος της εργασίας είναι φανερό ότι συμπεριλάβαμε ότι χρειάζεται για μια αλυσίδα γυμναστηρίων για να λειτουργήσει με οργάνωση.  
  
  
**Οργάνωση project**

Αρχικά σαν πρώτό βήμα ήταν να δημιουργήσουμε το ERD. Επειδή ο καθένας είχε διαφορετικές ιδέες αποφασίσαμε να φτιάξουμε από ένα ERD ο καθένας και να δουλεύαμε σε αυτό με την πιο καλή οργάνωση, χωρίς όμως να απορρίπταμε πλήρως το άλλο, μιας και οι 2 συνεισφέραμε στις ιδέες. Με την υλοποίηση του ERD χωρίσαμε την δουλεία ως εξής:

Ο Κοτσώνης(1066527) ανέλαβε το Relational model και να δημιουργήσει την βάση σε SQL ενώ ο Παπαδημητρίου(1066532) την γραφική εφαρμογή και την δημιουργία των Queries .

Στο τέλος θα γινόταν ένα merge των αρχείων κώδικα και θα είχαμε το τελικό αποτέλεσμα και μετά από κοινού την αναφορά. Δυστυχώς ο Κοτσώνης έφτιαξε μια σταθερή βάση σε αρχείο .sql, το οποίο δεν μπορούσε να παράγει μεγάλη ποσότητα τυχαίων δεδομένων (Με τους κανόνες που είχαμε θέσει εκ κοινού στο ERD), προκειμένου να εξετάσουμε πλήρως αναζήτηση πληροφοριών και την ταχύτητα των queries.

Επειδή ο χρόνος που είχε απομείνει δεν βόλευε και ο Παπαδημητρίου είχε ήδη εξοικειωθεί με την βιβλιοθήκη ανέλαβε εξ ολοκλήρου αυτό το κομμάτι, με τον Κοτσώνη να φτιάχνει εξ ολοκλήρου την παρουσίαση και την αναφορά.

**GitHub link to the project:**

<https://github.com/Apo-P/GymDatabase>