Βάσεις Δεδομένων : Εργασία Εξαμήνου

Δίκτυο διαχείρησης δανειστικών βιβλιοθηκών

Αλέξανδρος Μικέλης , 1066442

[up1066442@upnet.gr](mailto:up1066442@upnet.gr)

Γιάννος Σκαρπέτης, 1066539

[up1066539@upnet.gr](mailto:up1066539@upnet.gr)

1. Introduction

Στα πλαίσια του μαθήματος Βάσεις Δεδομένων του τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών του Πανεπιστημίου Πατρών μας δόθηκε εργασία εξαμήνου(7ο εξαμήνου) με θέμα «*Δημιουργία Βάσης Δεδομένων για δίκτυο διαχείρισης δανειστικών βιβλιοθηκών*».Το συγκεκριμένο θέμα αγγίζει πολλές πτυχές του μαθήματος και σκοπός της εργασίας είναι να κατανοήσουμε τις βασικές και πολυπλοκές έννοιες των βάσεων δεδομένων,να τις εφαρμόσουμε και να τις παρατηρήσουμε.

* 1. Περίληψη

Η βασική ιδέα της εργασίας είναι να φτιάξουμε μια βάση δεδομένων για ένα σύστημα/δίκτυο με πολλές βιβλιοθήκες οι όποιες θα δανείζουν βιβλία , θα δανείζονται βιβλία απο άλλες βιβλιοθήκες του δικτύου , θα διαχειρίζονται πελάτες και εργαζόμενους και θα συντηρούν τα βιβλία τους.Κάνοντας μια οικονομική προσέγγιση του θέματος καταλήξαμε ότι η βιβλιοθήκη θα χρεώνει μέσω memberships τον δανεισμό περισσότερων από 2 βιβλίων μηνιαία.

Αρκετό ενδιαφέρον έχει η διαδικασία του Inter Library Loaning ,με την οποία δίνουμε την δυνατότητα στις βιβλιοθήκες του συστήματος αν δεν έχουν κάποιο βιβλίο που θα τους ζητηθεί να μπορούν να το δανειστούν απο άλλη βιβλιοθήκη του συστήματος.Ως προς το υλοποιητικό κομμάτι φτιάξαμε μια εσωτερική εφαρμογή που σε πραγματικές καταστάσεις θα την χρησιμοποιούν οι εργαζόμενοι των βιβλιοθηκών για να κάνουν ενέργειες ως προς την βιβλιοθήκη εργασίας τους και οι διαχειριστές της εφαρμογής με σκοπό να κάνουν ενέργειες ως προς τις βιβλιοθήκες του συστήματος.

* 1. Μεθοδολογία

Η πορεία σκέψη μας ήταν αυτή όπου έχουμε αναπτύξει και στα πλαίσια του μαθήματος πάνω σε παρόμοια αλλά πιο απλά προβλήματα. Ξεκινήσαμε προσπαθώντας να οριοθετήσουμε τη βάση μας και συντάξαμε έτσι τον μικρόκοσμο του προβλήματος.Ύστερα ακολούθησε το Entity Relational Design (ERD) όπου και ασχοληθήκαμε διεξοδικά για να βεβαιωθούμε ότι όλες οι οντότητες έχουν οριστεί σωστά και περιέχουν τις απαραίτητες συμπεριφορές και συνεχίσαμε στο Relational Schema.Πλέον δίνουμε ενδιαφέρον στον σωστό ορισμό των κύριων (primary key) και ξένων κλειδιών (foreign key). Αφού ολοκληρώθηκε το Λογικό Μοντέλο της βάσης ,μέσα απο το DB Browser for SQLite ξεκινήσαμε να την δομούμε σύμφωνα με τα προυγούμενα μοντέλα που φτιάξαμε. Εν συνεχεία προχωρήσαμε στην εισαγωγή δεδομένων στη βάση με σκοπό να εξετάσουμε αν δομήθηκε επιτυχώς και σύμφωνα με τις προδιαγραφές της.Πρώτα λοιπόν με απλές CRUD εντολές (Create/Read/Update/Delete) και ύστερα με ερωτήματα τυπικών αναζητήσεων όπου δημιουργήσαμε, ολοκληρώσαμε το κομμάτι του testing.Εν τέλει προγραμματίσαμε μια εφαρμογή όπου προορίζεται για τον εργαζόμενο και κάνει δυναμικά ερωτήματα στη βάση, σύμφωνα με τα ερωτήματα του χρήστη εργαζόμενου.

1. Αναλυση μεθοδολογιας

Θα αναλύσουμε πιο συγκεκριμένα τις ενέργειες που έγιναν στην περίληψη της μεθοδολογίας [**1.2**]. Με σκοπό την καλύτερη συνεργασία μας χρησιμοποιήσαμε τεχνολογίες collaboration , και ειδικότερα για τον πηγαίο κώδικα της εργασίας χρησιμοποιήσαμε **github** και git **fork** και για την αποθήκευση των sql queries εκμεταλλευτήκαμε την **popSQL**.

* 1. Μικρόκοσμος

Διαπραγματευόμαστε ένα σύστημα το οποίο διαχειρίζεται πολλές βιβλιοθήκες όπου η κάθε μία έχει ένα όνομα, μία διεύθυνση και μία τοποθεσία. Κάθε βιβλιοθήκη αποτελείται από τομείς, κάθε τομέας χαρακτηρίζεται από ένα όνομα, έναν αριθμό και σε ποιον όροφο της βιβλιοθήκης βρίσκεται. Για τη διευκόλυνση κάθε βιβλιοθήκης χωρίζουμε τα βιβλία όπου υπάρχουν σε κατηγορίες. Για την ομαλή οργάνωση της βιβλιοθήκης τοποθετούνται τα βιβλία σε συγκεκριμένες θέσεις (Διαδρόμος, Ράφι). Επίσης, υπάρχει μία ειδική καταγραφή για τα βιβλία τα οποία θέλουν συντήρηση.

Κάθε βιβλιοθήκη εξυπηρετεί πελάτες όπου έρχονται σε αυτή με σκοπό να δανειστούν βιβλία. Δίνουμε την επιλογή στο πελάτη να επιλέξει μεταξύ κάποιων συνδρομών (Επί πληρωμή συνδρομή, Free …) ώστε να έχει τη δυνατότητα να δανειστεί βιβλία. Κάθε πελάτης μπορεί να δανειστεί μέχρι κάποιο αριθμό βιβλίων ανάλογα με τη συνδρομή. Ο πελάτης έχει μια συγκεκριμένη ημερομηνία που πρέπει να τα επιστρέψει. Στη περίπτωση όπου ο πελάτης θέλει να δανειστεί ένα βιβλίο το οποίο δεν είναι διαθέσιμο μπορεί να το παραγγείλει και μπαίνει σε λίστα αναμονής.

Κάθε βιβλιοθήκη έχει κάποιους εργαζόμενους που έχουν ένα συμβόλαιο με τη βιβλιοθήκη το οποίο ορίζει την εργασία τους(Τομέας εργασίας, προκαθορισμένο ποσό πληρωμής, ημερομηνία κ.λ.π). Επίσης βάσει του συμβολαίου αυτού σε συνδυασμό με έξτρα ώρες εργασίας, μπόνους κ.λ.π βγαίνει και η τελική μηνιαία πληρωμή του εργαζόμενου. Υπάρχουν προϊστάμενοι όπου είναι υπεύθυνοι για μία ομάδα – τομέα. Επίσης όποιος είναι μάγκας δεν παίρνει άδεια αλλά γι’αυτούς που είναι τεμπέληδες υπάρχουν κάποιες μέρες όπου μπορούν να πάρουν άδεια, τις οποίες καταγράφουμε.

Το δίκτύο μας θέλουμε να υποστηρίζει InterLibrary Loan , δηλαδή αν μια βιβλιοθήκη δεν έχει κάποιο βίβλιο να μπορεί να δανειστεί από κάποια άλλη βιβλιοθήκη το εκάστοτε βιβλίο.

* 1. ERD (Entity Relational Design)

Για την δημιουργία του διαγράμματος οντοτήτων χρησιμοποιήσαμε το [ERDMaker](https://erdmaker.com/) .Ξεκινώντας από το Root Entity ,όπου είναι το Library, πρέπει να καταγράφουμε το όνομα της βιβλιοθήκης,την ακριβή διεύθυνση ,τη γενική τοποθεσία της και το τηλέφωνό της.Η βιβλιοθήκη έχει από 1 έως πολλά Section.Το Section χαρακτηρίζεται από ένα όνομα ,ένα id , ένα νούμερο και έναν όροφο στον οποίο ανήκει.Επίσης σε ένα Section περιέχoνται από 0 έως πολλές κατηγορίες , όπου το 0 υπάρχει με την προοπτική ότι στη βιβλιοθήκη μπορεί να υπάρξουν Sections ,όπως για παράδειγμα τμήμα υπολογιστών ,που να μην περιέχουν καμία κατηγορία βιβλίου.Η Κατηγορία χαρακτηρίζεται μόνο από το όνομα και id .Για λόγους απλοποίησης και ρεαλιστικότητας τοποθετούμε τις κατηγορίες σε συγκεκριμένες θέσεις όπου διακρίνονται από διάδρομο και ράφι.

Για μια βιβλιοθήκη καταγράφουμε ακριβώς πόσα βιβλία έχει γενικά .Το βιβλίο αποτελείται από τον τίτλο, τον συγγραφέα , τον εκδότη, σε τι κατάσταση είναι , το isbn δηλαδή τον χαρακτηριστικό αριθμό για βιβλία και το ID του.Καθένα βιβλίο ανήκει και σε μια κατηγορία, ενώ έχουμε κατηγορίες με πολλά βιβλία.Επίσης η βιβλιοθήκη εξυπηρετεί πολλούς πελάτες. Ο πελάτης χαρακτηρίζεται από όνομα , επίθετο,τηλέφωνο,email,διεύθυνση και ένα id.Ένας πελάτης πληρώνει μια συνδρομή , η οποία αποτελείται από την αρχή και το τέλος της συνδρομής σε ημερομηνίες, το αριθμό των βιβλίων που δικαιούται να δανειστεί ανά μήνα, το ποσό που πληρώνει ο πελάτης ανά μήνα και τον τίτλο της συνδρομής (gold,silver,free).Ένας πελάτης μπορεί να δανειστεί πολλές φορές ανάλογα με την συνδρομή του και πρέπει να καταγράφουμε για τον δανεισμό την ημερομήνια που το πήρε , την ημερομηνία που το επέστρεψε και την ημερομήνια που κανονικά είναι υποχρεωμένος να το επιστρέψει.Επίσης ένας πελάτης αν θέλει να δανειστεί ένα βιβλίο που δεν υπάρχει μπορεί να το παραγγείλει.Μετά απο κάθε επιστροφή βιβλίου το βιβλίο μπορεί να αλλάξει condition , αν το condition γίνει ‘Bad’ πρέπει να μπει σε διαδικασία συντήρησης όπου και το καταγράφουμε .Η συντήρηση χαρακτηρίζεται από τον τύπο συντήρησης ,την ημερομηνία έναρξης και την ημερομηνία λήξης συντήρησης.

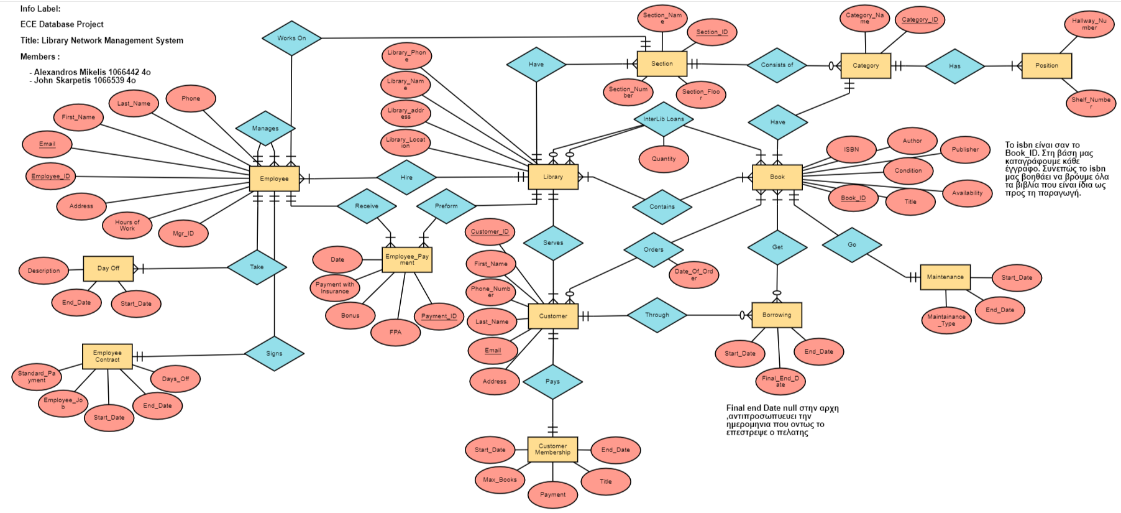
Η βιβλιοθήκη προσλαμβάνει υπαλλήλους όπου δουλεύουν σε sections.Επιπρόσθετα ο υπάλληλος χαρακτηρίζεται απο όνομα, επώνυμο, τηλέφωνο, email, διεύθυνση, ώρες δουλειάς, id και manager id. Επειδή ένας από τους εργαζόμενους είναι και διευθυντής τμήματος(Section) έχουμε αυτοσυσχέτιση στην οντότητα employee.Εν συνεχεία ένας εργαζόμενος δικαιούται κάποιες μέρες ρεπό, όπου και καταγράφουμε τις ημερομηνίες και την αιτία. Εν τέλει όσον αφορά τον εργαζόμενο, όταν προσλαμβάνεται πρέπει να υπογράφει ένα συμβόλαιο με χαρακτηριστικά τις ημερομηνίες έναρξης και λήξης συμβολαίου, πόσες μέρες ρεπό δικαιούται και την πληρωμή που δικαιούται χωρίς ασφάλιση. Η βιβλίοθηκη πληρώνει τους εργαζόμενους με ασφάλιση,ίσως κάποιο bonus και εμείς καταγράφουμε και την ημερομηνία της πληρωμής αυτής.Εν κατακλείδι όσο αφορά το Inter library loaning το πετυχαίνουμε με μια τριαδική συσχέτιση ανάμεσα στη βιβλιοθήκη , τη βιβλιοθήκη και το βιβλίο για να μπορούμε να καταγράφουμε πότε μια βιβλιοθήκη (getter) ζητάει ένα βιβλίο από μια άλλη (sender).

Figure 1: Project ERD ([ERDMaker](https://erdmaker.com/))

* 1. Relational Schema

Για την δημιουργία του λογικού μοντέλου χρησιμοποιήσαμε το [DBDesigner](https://www.dbdesigner.net/) . Σχεδιάσαμε ακολουθώντας τους κανόνες μετασχηματισμού από ERD σε Relational όπου μας εξασφαλίζουν τον σωστό ορισμό των κλειδιών μας.Οι κανόνες που ακολουθήσαμε είναι οι εξής :

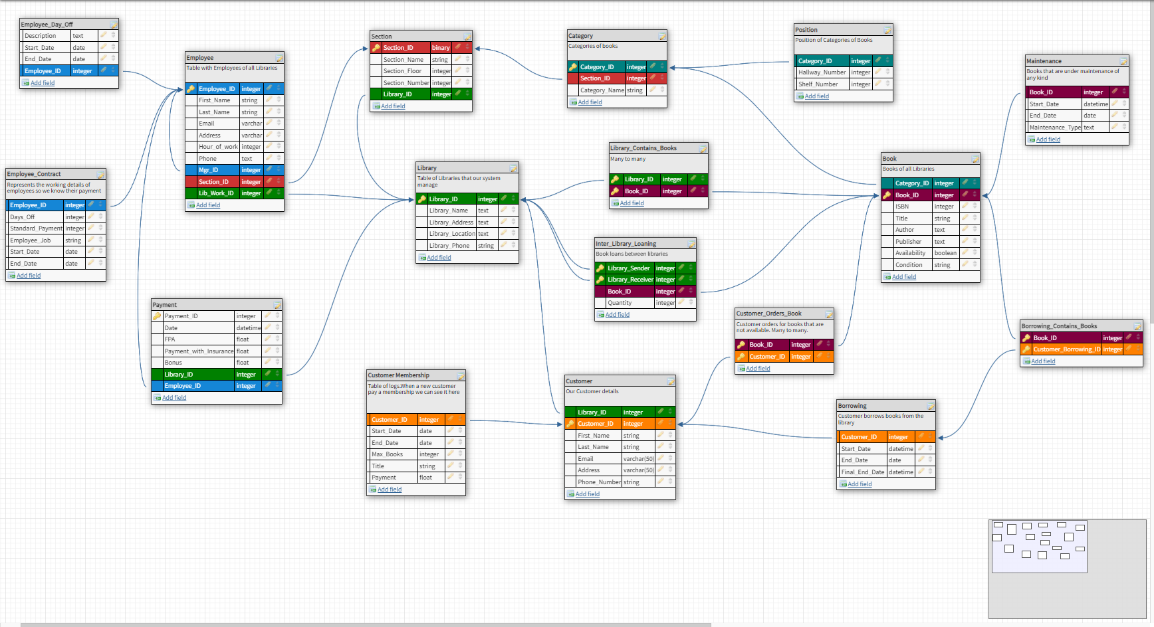
* Συσχετίσεις 1-1 : Το ξένο κλειδί πάει στον πίνακα με τα περισσότερα στιγμιότυπα
* Συσχετίσεις 1-Μ : Το ξένο κλειδί εμφανίζεται στην πλευρά του M
* Συσχετίσεις Μ-Μ : Τα κλειδιά τους φτιάχνουν το σύνθετο κλειδί του τρίτου βοηθητικού πίνακα

Figure 2: Project Conceptual Design ([DBDesigner](https://www.dbdesigner.net/))

* 1. SQLite

*2.4.1 Creation Of Database*

Για να φτιάξουμε τη βάση χρησιμοποιήσαμε το περιβάλλον του DB Browser for SQLite.Η βάση δημιουργήθηκε σύμφωνα με το Relational Schema . Δόθηκε ιδιαίτερη προσόχη στη σύνδεση των κλειδιών και των εννοιών, και αφού δημιουργήθηκε εξ ολοκλήρου η βάση έγινε έρευνα ως προς το σενάριο ανανέωσης και διαγραφής ενός κύριου κλειδιού(ON UPDATE, ON DELETE).Το περιβάλλον προσφέρει γραφική διεπαφή που κάνει πιο εύκολη τη διαδικασία της υλοποίησης.

Text

Description automatically generatedFigure 3: Database (DB Browser for SQLite)

*2.4.2 CRUD Εντολές*

Οι εντολές CRUD (Create/Read/Update/Delete) είναι η ακόλουθη αντιστοίχιση:

* Create 🡪 INSERT
* Read 🡪 SELECT
* Update 🡪 UPDATE
* Delete 🡪 DELETE

Graphical user interface, text, application

Description automatically generatedΑρχικά χρησιμοποιήσαμε την εντολή INSERT για να αρχίσουμε να βάζουμε δεδομένα στη βάση μας , με σκοπό να προχωρήσουμε στο 2.4.3 και να δουλέψουμε παράλληλα. Μέσα στο αρχείο CRUD.sql θα βρείτε στην αρχή όλα τα INSERT που κάναμε για να γεμίσουμε τη βάση μας με δεδομένα .Tα δεδομένα αυτά βρέθηκαν από το ιντερνετ και κάποια είναι dummy variables, για να μπορούμε να διεξάγουμε test.Ύστερα για τις υπόλοιπες CRUD εντολές χρησιμοποιήσαμε σενάρια που μπορούν να γίνουν στην πραγματικότητα και χρειάζονται Read και Update .Ένα απο αυτά τα σενάρια φαίνονται στο **Figure 4**. Όλα τα σενάρια μπορούν να παρατηρηθούν πιο διεξοδικά και μέσα απο το αρχείο των CRUD εντολών. Το κομμάτι του DELETE το διαπραγματευτήκαμε φτιάχνοντας βιβλία με λαθος δεδομένα όποτε έπρεπε και να σβηστούν

Figure 4: Lending a Book Scenario

*2.4.3 Τυπικές Αναζητήσεις*

Αφού ολοκληρώσαμε την ενσωμάτωση δεδομένων στη βάση μας, φτιάξαμε ερωτήματα τυπικών αναζητήσεων παρόμοια με τα πλαίσια του εργαστηρίου του μαθημάτος με σκοπό να αλληλεπιδράσουμε με τη βάση δεδομένων και να την επαληθέυσουμε. Στην προσπάθειά μας αυτή δημιουργήθηκαν 38 ερωτήματα απλά και πολύπλοκα, τα οποία αποτελούν test cases για το αν η βάση μας λειτουργεί ικανοποιητικά.Πιο εκτεταμένα όλα τα ερωτήματα μαζί με τις απαντήσεις (queries) βρίσκονται στο αρχείο TestingQueries.sql. Μέσα από τη συγκεκριμένη διαδικάσία καταλάβαμε προβλήματα της βάσης που μας ανάγκασαν να την ξαναδομήσουμε αλλάζοντας ταυτόχρονα και τα μοντέλα μας.

Text

Description automatically generated Figure 5 : Some of Testing Questions and Queries

* 1. Python

Η ανάπτυξη της εφαρμογής έγινε μέσα από την γλώσσα προγραμματισμού python.Χρησιμοποιήσαμε βιβλιοθήκες για GUI (streamlit) , για Data Visualazation (matplotlib) και για τη σύνδεση και αλληλεπίδραση με τη βάση την sqlite3.Η εφαρμογή αντιπροσωπεύει ένα εσωτερικό σύστημα που θα δίνουμε σε κάθε βιβλιοθήκη του συστήματός μας για να μπορεί να διαχειρίζεται ο εργαζόμενος τις ανάγκες της καθημερινότητας.

Ο εργαζόμενος θα πρέπει να συνδέεται με το email του και το ΑΦΜ του και το σύστημά μας ανάλογα με τα permissions που πρέπει να έχει του δίνει τη δυνατότητα να κάνει κάποιες ενέργειες.Συγκεκριμένα το σύστημά μας δέχεται δύο τύπους εργαζομένων και ανάλογα δίνει δικαιώματα, δεχόμαστε έναν τυπικό εργαζόμενο που πρέπει να δανείσει βιβλία ή που πρέπει να τσεκάρει ποιοι πελάτες δεν έχουν επιστρέψει κάποιο βιβλίο εγκαίρως, καθώς και έναν Admin όπου θα ειναι αυτός που κάνει αρχικοποίηση των βασικών δεδομένων όταν ένας καινούργιος πελάτης/βιβλιοθήκη μπει στο σύστημα μας.

Ως προς το κομμάτι των ενεργειών για τον τυπικό εργαζόμενο έχουμε δώσει τη δυνατότητα να βλέπει την κατάσταση όλων των βιβλιών ή και συγκεκριμένων ανάλογα με κάποιο φιλτραρίσματα που θα δώσει ο ίδιος, να μπορεί να βάζει ένα καινούργιο βιβλίο στην βιβλιοθήκη που δουλεύει , να μπορεί να κάνει update και delete ένα βιβλίο , να μπορεί να δανείσει ένα βιβλίο , να μπορεί να κάνει αίτημα δανεισμού σε μια άλλη βιβλιοθήκη,να μπορεί να καταχωρήσει μια παραγγελία για ένα βιβλίο και να δώσει ένα πρόστιμο στους πελάτες όπου έχουν αργήσει να επιστρέψουν ένα ή πολλά βιβλία.

Από την άλλη ο διαχειριστής μπορεί να κάνει αυτά και ακόμα να διαχειρίζεται βιβλιοθήκες μέσα στη βάση , να διαχειρίζεται εργαζόμενους και τμήματα βιβλιοθηκών.Η ανάπτυξη μιας ολοκληρωμένης εφαρμογής στα πλαίσια του εξαμήνου είναι ανέφτικη και παρά το μικρό χρονικό διάστημα οι πιο βασικές λειτουργίες του δικτύου μας έχουν καλυφθεί από την εφαρμογή.Επειδή ο κώδικας είναι αρκετά εκτενής τον έχουμε εμπλουτίσει με αρκετά επεξηγηματικά σχόλια σε όλη την έκταση σου όπου μπορείτε να παρατηρήσετε και να κατανοήσετε μέσα από τα αρχεία .py του συγκεκριμένου Project.

Graphical user interface

Description automatically generated

Figure 6 : Simple Employee Login in our App

1. Αξιολογηση project
   1. Τρόποι αξιολόγησης

Αφού φτιάξαμε τη βάση και της προσθέσαμε δεδομένα για να αξιολογήσουμε πόσο καλά ανταπεξέρχεται σε ενέργειες που πρέπει να γίνουν και γενικότερα στην διαχείριση των δεδομένων δημιουργήσαμε ερωτήματα τυπικών αναζητήσεων ώστε να δούμε αν τα δεδομένα που επιστρέφονται είναι σωστά.Τρόπος αξιολόγησης είναι η δικιά μας άποψη επί της βάσης και της επιστροφής των δεδομένων από τις τυπικές αναζητήσεις, δηλαδή πρέπει χειροκίνητα να δουμε αν τα δεδομένα που ζητήσαμε είναι τα δεδομένα που επιστράφηκαν κάτι που είναι εφικτό μέσα από το περιβάλλον της SQLite. Εν τέλει ακόμα ένας τρόπος αξιολόγησης είναι να δούμε το σενάριο του update και delete στα κύρια κλειδιά των table μας,δηλάδη τι θα συμβεί στο foreign key όταν το primary key μας αλλάξει ή διαγραφεί.

Χρησιμοποιήσαμε λοιπόν στον ορισμό της βάσης, τη λειτουργία ON UPDATE CASCADE , ON DELETE CASCADE, ON DELETE SET NULL.Καταλήξαμε ότι για το Root Entity όπου είναι η βιβλιοθήκη η διαδικασία ON DELETE CASCADE είναι αρκετά επίφοβη γι’ αυτό και την αγνοήσαμε.Δοκιμάσαμε επίσης ηθελημένα να βάλουμε λάθος δεδομένα με σκοπό να μας επιστραφεί error.Εν κατακλείδι κύριως τρόπος αξιολόγησης της βάσης δεδομένων για το δίκτυο βιβλιοθηκών είναι τα Test cases από τις τυπικές αναζητήσεις και οι CRUD εντολές σε συνδυασμό με τη δική μας αίσθηση για το πως πρέπει να δουλεύει η βάση.

* 1. Κριτήρια αξιολόγησης

Τα κριτήρια αξιολόγησης διεξάγονται μέσα από την εφαρμογή των τρόπων αξιολόγησης που αναφέραμε στην 3.1. Ξεκινόντας με τις τυπικές αναζητήσεις όπου δημιουργήθηκαν στο 2.4.3 , η εγκυρότητα των αποτελεσμάτων που επιστρέφουν θα αποτελέσει ένα κριτήριο.Για τα συγκεκριμένα 38 queries τα αποτελέσματα που επιστράφηκαν ήταν έγκυρα.Η επαλήθευση έγινε με την σύγκριση των αποτελέσματων που επιστράφηκαν και τα πραγματικά δεδομένα της βάσης, όπου έγινε εφικτό μέσα από το Data view του SQLite Browser.Μέσω αυτού του κριτηρίου ανακαλύψαμε ότι η βάση μας σε συγκεκριμένες ενέργειες υστερεί στην αποδοτικότητα. Υπάρχει ίσως μια καλύτερη δομή της βάσης όπου κάποιες ενέργειες θα γινόνταν πιο εύκολες.Μια τέτοια ενέργεια είναι η αναζήτηση όλων των βιβλίων που τα έχουν δανειστεί.

Δοκιμάσαμε να διαγράψουμε/ανανεώσουμε primary keys σε κλειδιά που έχουν σαν ιδιότητα ON DELETE/UPDATE CASCADE . Είδαμε το προβλεπόμενο που ήταν να ανανεώνονται/διαγράφονται και τα ξένα κλειδιά που είναι συνεδεμένα. Σημειωτέων ότι δεν πραγματοποιήσαμε πολλά test cases όσο αφορά το κομμάτι της διαγραφής παρα μόνα όσα χρειάζονταν για να αποτελέσουν ένα καλό κριτήριο αξιολόγησης.Επίσης αντίστοιχα αποτελέσματα είχαμε και στο σενάριο του ON DELETE SET NULL ,όπου για παράδειγμα στη συσχέτιση SECTION-CATEGORY δεν θέλουμε να διαγράφονται και όλες οι κατηγορίες ενός section αν αυτό διαγραφεί.

Τελευταίο κριτήριο αξιολόγησης αποτέλεσε η επιτηδευμένη απόπειρα να εισάγουμε λάθος δεδομένα με σκοπό να λάβουμε error .Αυτό μας βοήθησε για ακόμη μια φορά να επαληθεύσουμε την επικοινωνία μεταξύ των table μας με βάση το πώς έχουν οριστεί απο εμάς.

1. Δεδομενα

Η εργασία όπως και η αξιολόγηση της βάσης όπως έχει αναφερθεί χρειάστηκε δεδομένα.Κάποια από αυτά τα δεδομένα τα λάβαμε από ιστοσελίδες όπως ο εύδοξος, το public (δεδομένα βιβλίων) και η google (δεδομένα βιβλιοθηκών) , και κάποια απο αυτά δημιουργήθηκαν μέσα απο εμάς όπως είναι τα στοιχεία των πελατών και των εργαζομένων των βιβλιοθηκών , οι κατηγορίες και τα τμήματα κάθε βιβλιοθήκης.Να διευκρινιστεί ότι τα δεδομένα που φτιάχτηκαν από εμάς δεν ανταπεξέρχονται σε πραγματικά δεδομένα διότι αγγίζουν ευαίσθητα προσωπικά δεδομένα (ΑΦΜ, email κλπ).Στην πραγματικότητα αυτά τα δεδομένα θα μπαίνουν encrypted στη βάση ειδικά όταν μιλάμε για δεδομένα κωδικών.

Τα δεδομένα ενσωματώνονται στη βάση μας με δύο τρόπους.Μέσα απο τις INSERT εντολές στο αρχείο crud.sql όπου το χρησιμοποιήσαμε σαν initialization για να μπορούμε να κάνουμε το testing της βάσης μέσα από τις τυπικές αναζητήσεις.Ο δεύτερος ,πιο ρεαλιστικός τρόπος ενσωμάτωσης δεδομένων στη βάση μας, είναι μέσα από την εφαρμόγη, όπου θα έχουμε καθημερινά interactions με πελάτες , βιβλιοθήκες , εργαζόμενους κλπ.

1. Ολοκληρωση εργασιασ
   1. Κύριες ενέργειες

Η εργασία από τη στιγμή που αποτελείται απο δύο ατόμα θα χρειαστεί μια τεχνολογία αλληλεπίδρασης.Ξεκινήσαμε λοιπόν φτιάχνοντας το δικό μας github repository όπου θα κάνουμε share τον κώδικά μας και θα μπορούμε να δουλεύουμε ασύγχρονα.Εν συνεχεία το phase A (Μικρόκοσμος,ERD,Conceptual Design) το ολοκληρώσαμε από τις πρώτες εβδομάδες σε συνεργατικό επίπεδο μέσα από δια ζώσης meetings.Στο phase B ξεκίνησε η παράλληλη δουλειά των μελών της ομάδας με αυτή να χωρίζεται με τέτοιο τρόπο ώστε να μην χαλάει η δουλειά του συνεργαζόμενου.Χωρίσαμε λοιπόν την δουλειά στη δημιουργία της βάσης (όπου έγινε συνεργατικά) και στη δημιουργία CRUD εντολών (1 άτομο) και δημιουργία τυπικών αναζητήσεων (1 άτομο). Το phase Γ και η έκθεση υλοποιηθήκε απο κοινού με ποσοστιαίες διαφορές για το κάθε μέλος αναλογικά.

* 1. Κύριες ενέργειες

Μικέλης Αλέξανδρος : Μικρόκοσμος, ERD, Conceptual Design, DB Creation (30%),CRUD, Python App(70%), Essay(50%)

Σκαρπέτης Ιωάννης : Μικρόκοσμος, ERD, Conceptual Design, DB Creation(70%),Typical Searches, Python App(30%), Essay (50%)

1. Χρονοδιαγραμμα

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Weeks | Dates | Μικρόκοσμος | ERD | Conceptual Design | DB Creation | CRUD | Typical Searches | Python | Essay | Comments |
| 4η | 5/11/2021 |  |  |  |  |  |  |  |  | 1ο Meeting : Αρχή Σύνταξης Μικρόκοσμου/ Έρευνα . |
|  | 8/11/2021 |  |  |  |  |  |  |  |  | 2ο Meeting : ERD |
| 5η | 12/11/2021 |  |  |  |  |  |  |  |  |  |
|  | 14/11/2021 |  |  |  |  |  |  |  |  | Updates Μικροκοσμος-ERD |
|  | 15/11/2021 |  |  |  |  |  |  |  |  | 3ο Meeting : ERD 🡪 Conceptual Design |
| 6η | 18/11/2021 |  |  |  |  |  |  |  |  |  |
|  | 20/11/2021 |  |  |  |  |  |  |  |  | Τελικός Μικρόκοσμος για 1η παρουσίαση |
|  | 22/11/2021 |  |  |  |  |  |  |  |  | Τελικό ERD για 1η παρουσίαση |
| 7η | 24/11/2021 |  |  |  |  |  |  |  |  | Πρώτη Παρουσίαση Project.Σχόλια καθηγήτων 🡪Βιβλιο συνδεση με Βιβλιοθήκη, Orders να γίνουν οντότητα |
|  | 27/11/2021 |  |  |  |  |  |  |  |  | Τελικά files ERD και Μικρόκοσμου σύμφωνα και με τις προτεινόμενες αλλαγές |
| 8η | 3/12/2021 |  |  |  |  |  |  |  |  |  |
|  | 4/12/2021 |  |  |  |  |  |  |  |  | Τελικό Conceptual |
| 9η | 11/12/2021 |  |  |  |  |  |  |  |  | Ερευνα/Περιβάλλον SQLite |
|  | 13/12/2021 |  |  |  |  |  |  |  |  |  |
| 10η | 17/12/2021 |  |  |  |  |  |  |  |  |  |
|  | 18/12/2021 |  |  |  |  |  |  |  |  |  |
|  | 19/12/2021 |  |  |  |  |  |  |  |  | Βασικά Inserts |
| 11η | 23/12/2021 |  |  |  |  |  |  |  |  |  |
|  | 24/12/2021 |  |  |  |  |  |  |  |  | Διορθώσεις Βάσης |
|  | 28/12/2021 |  |  |  |  |  |  |  |  |  |
|  | 30/12/2021 |  |  |  |  |  |  |  |  |  |
|  | 2/1/2022 |  |  |  |  |  |  |  |  |  |
|  | 3/1/2022 |  |  |  |  |  |  |  |  |  |
|  | 5/1/2022 |  |  |  |  |  |  |  |  | Διορθώσεις Βάσης |
|  | 9/1/2022 |  |  |  |  |  |  |  |  |  |
| 12η | 12/1/2022 |  |  |  |  |  |  |  |  |  |
|  | 13/1/2022 |  |  |  |  |  |  |  |  | Τελικές Διορθώσεις Βάσης |
|  | 14/1/2022 |  |  |  |  |  |  |  |  | Ένταξη επιπλέον δεδομένων |
|  | 15/1/2022 |  |  |  |  |  |  |  |  | Ολοκλήρωση Έκθεσης |
|  | 16/1/2022 |  |  |  |  |  |  |  |  | Παράδοση και Double Check εργασίας |
| 13η | 19/1/2022 |  |  |  |  |  |  |  |  | Παρουσίαση Εργασίας |

1. Οδηγίεσ Εγκατάστασης και παραδειγματα χρησησ

Αρχικά μπορείτε να κατεβάσετε όλοκληρο το περιεχόμενο που αναπτύχθηκε για την εργασία [εδω](https://github.com/AlexandrosMikelis/ECE_DB_SemesterProject) . Μαζί με το περιεχόμενο θα βρείτε στο README.md οδηγίες εγκατάστασης και μια μικρή εισαγωγή της εργασίας.Παρόλα αυτά ακριβής διαδικασία για να μπορείτε να τρέξετε τον κώδικα είναι :

1. Κατεβάστε το zip αρχείο του πηγαίου κώδικα από το github.
2. Κάντε το extract σε ένα τοπικό directory.
3. Ανοίξτε ένα terminal(cmd) στο directory που αποθηκεύσατε και εκτελέστε την εντολή pip install -r requirements.txt .
4. Αν δεν σας αναγνωρίζει το pip τότε υπάρχουν δύο σενάρια αντιμετώπισης , αν δεν έχετε εγκατεστημένη την python τότε εγκαταστήστε την μέσα από [εδώ](https://phoenixnap.com/kb/how-to-install-python-3-windows) και βεβαιωθείτε ότι κάνετε κλικ Add to path διαφορετικά αν έχετε εγκατεστημένη την python αλλά δεν την έχετε βάλει στο path ακολουθήστε τις οδηγίες [εδώ](https://datatofish.com/add-python-to-windows-path/).
5. Τρέξτε στο terminal την εντολή streamlit run App.py
6. Ένα καινούργιο window θα σας ανοιξει ένα page στο internet browser σας και πλέον είστε στην εφαρμογή

Στην εφαρμογή μπορείτε να συνδεθείτε με τα παρακάτω στοιχεία σαν ένας απλός εργαζόμενος της βιβλιοθήκης :

Email : [JP@gmail.com](mailto:JP@gmail.com)

SSN : 23080002895

Και μπορείτε πλέον να αλληλεπιδράσετε με τη βάση μέσω της εφαρμογής.Αν θέλετε να ανοίξετε και να τρέξετε τα queries ή να αλληλεπιδράσετε με τη βάση σε περιβάλλον SQL παρακαλούμε να ακολουθήσετε τα βήματα [εδώ](https://eclass.upatras.gr/modules/document/file.php/EE766/%CE%91%CF%83%CE%BA%CE%AE%CF%83%CE%B5%CE%B9%CF%82%20%CE%B5%CF%81%CE%B3%CE%B1%CF%83%CF%84%CE%B7%CF%81%CE%AF%CE%BF%CF%85/%CE%AC%CF%83%CE%BA%CE%B7%CF%83%CE%B7%206.pdf).

Graphical user interface

Description automatically generatedΠαραδείγματα της εφαρμογής βλέπουμε παρακάτω.

Graphical user interface

Description automatically generated with medium confidenceGraphical user interface, application

Description automatically generated