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 διαχείρησης δανειστικών βιβλιοθηκών*».Το συγκεκριμένο θέμα αγγίζει πολλές πτυχές του μαθήματος και σκοπός της εργασίας είναι να κατανοήσουμε βασικές και πολυπλοκές έννοιες των βασεων 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 , δανείζονται βιβλία απο άλλες βιβλιοθήκες του δικτύου , διαχειρίζονται πελάτες και εργαζόμενους και θα συντηρούν τα βιβλία τους.Κάνοντας μια οικονομική προσέγγιση του θέματος καταλήξαμε οτι οι βιβλιοθήκη θα χρεώνει μέσω memberships τον δανεισμό περισσοτέρων απο 2 βιβλίων μηνιαια.

Αρκέτα ενδιαφέρον έχει η διαδικασία του Inter Library Loaning ,να δέινουμε την δυνατότητα στις βιβλιοθήκες του συστήματος αν δεν έχουν κάποιο βιβλίο που τους ζητηθεί να μπορούν να το δανείστουν 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 διαχειρηστές της εφαρμογής με σκοπό να κάνουν ενέργειες ως προς τις βιβλιοθήκες του συστήματος.

* 1. Μεθοδολογία

Η πορεία σκέψη μας ήταν αυτή όπου έχουμε αναπτύξη και στα πλαίσια του μαθήματος πάνω σε παρόμοια αλλα πιο απλα προβλήματα. Ξεκινήσαμε προσπαθώντας να οριοθετήσουμε την 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 στην 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 τέλη προγραμματίσαμε μια εφαρμογή όπου προορίζεται για τον εργαζόμενο και κάνει δυναμικά ερωτήματα στην 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 , πιο συγκρεκριμένα χρησιμοποιήσαμε για τον κώδικα **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 διέυθυνση 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 ποσό 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 κάποιο συμβόλαιο με τη βιβλιοθήκη το οποίο ορίζει την εργασία τους(Τομέας εργασίας, προκαθορισμένο ποσό πληρωμής, ημερομηνία κ.λ.π). Επίσης βάση του συμβολαίου αυτού σε συνδυασμό με έξτρα ώρες εργασίας, μπόνους κ.λ.π βγαίνει και η τελική μηνιαία πληρωμή του εργαζόμενου. Υπάρχουν προϊστάμενοι οπού 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 δικτυο μας θέλουμε να υποστηρίζει InterLibrary Loan , δηλαδή αν μια βιβλιοθήκη δεν έχει κάποιο βιβλιό να μπορεί να δανειστεί απο κάποια άλλη βιβλιοθήκη το εκάστοτε βιβλίο.

* 1. ERD (Entity Relational Design)

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

Για μια βιβλιοθήκη καταγράφουμε ακρίβως πόσα βιβλία έχει γενικά .Το βιβλίο αποτελείται απο τον τίτλο, τον συγγραφεα , τον εκδότη, σε τι κατάσταση είναι , το isbn ένα χαρακτηριστικό αριθμό για βιβλία και το ID του.Κάθε ένα βιβλίο ανήκει και σε μια κατηγορία ενώ έχουμε κατηγορίες με πολλά βιβλία.Επίσης η βιβλιοθήκη εξυπηρετεί πολλούς πελάτες. Ο πελάτης χαρακτηρίζεται απο όνομα , επίθετο,τηλέφωνο,email,διευθυνση και ένα id.Ένας πελάτης πληρώνει μια συνδρομή , η οποία αποτελείται απο την αρχή και το τέλος της συνδρομής σε ημερομηνίες, το πόσα βιβλία δικαιούται να δανειστεί ανα μήνα, το πόσο πληρώνει ο πελάτης ανα μήνα και τον τίτλο της συνδρομής (gold,silver,free).Ένας πελάτης μπορεί να δανείστεί 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 μπεί σε διαδικασία συντήρησης όπου και το καταγράφουμε .Η συντήρηση χαρακτηρίζεται απο τύπο συντήρησης ,ημερομήνία έναρξης και ημερομηνία λήξης συντήρησης.

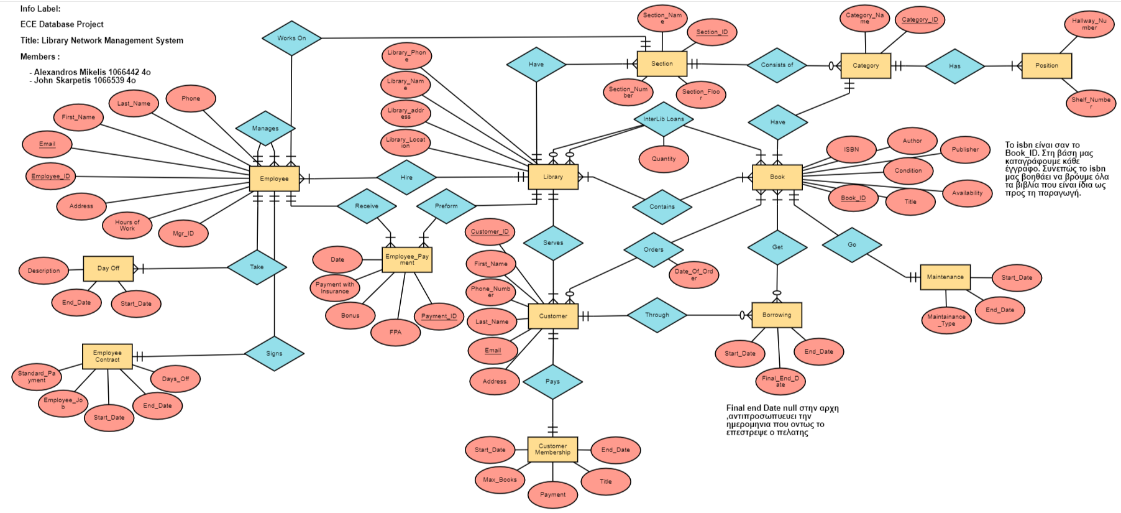
Η βιβλιοθήκη προσλαμβάνει υπαλλήλους όπου δουλέυουν σε sections.Επιπρόσθετα ο υπάλληλος χαρακτηρίζεται απο όνομα,επώνυμο,τηλέφωνο,email,διεύθυνση,ώρες δουλειάς,id και manager id.Επειδή ένας απο τους εργαζόμενους έιναι και διευθυντής τμήματος(Section) έχουμε αυτοσυσχέτιση στην οντότητα employee.Εν συνεχεία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

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

* 1. Relational Schema

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

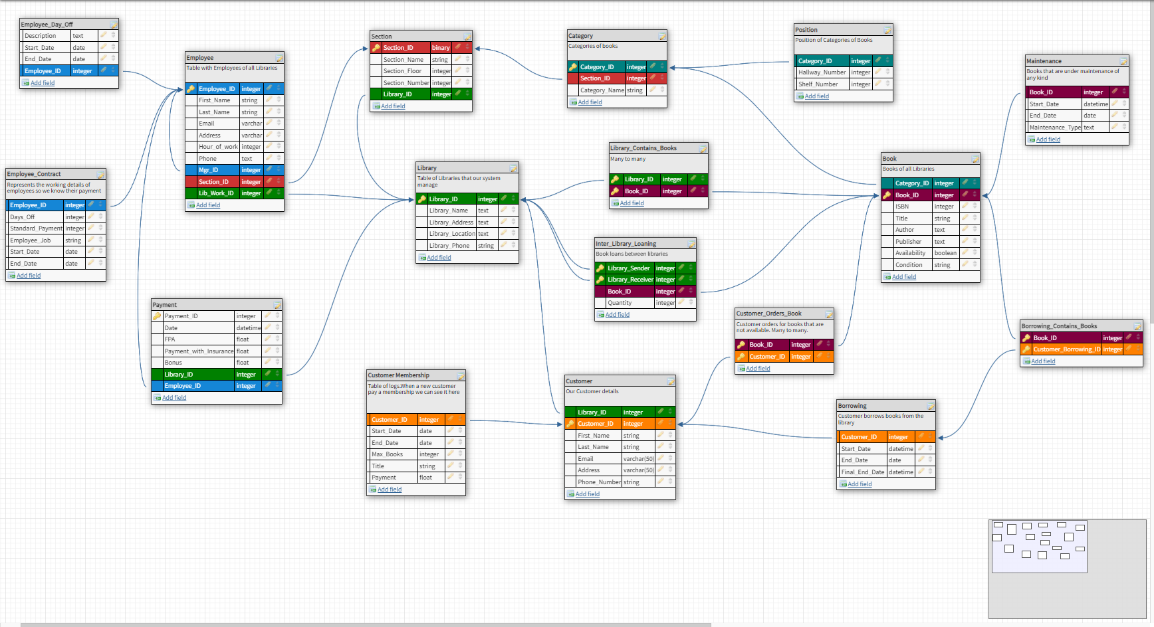
* Συσχετίσεις 1-1 : Το ξένο κλειδί πάει στον πίνακα με τα περισσότερα στιγμυότυπα
* Συσχετίσεις 1-Μ : Το ξένο κλειδί εμφανίζεται στην πλευρά του M
* Συσχετίσεις Μ-Μ : Τα κλειδία τους φτιάχνουν το σύνθετο κλειδί του τρίτου βοηθητικό πίνακα

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

* 1. SQLite

*2.4.1 Creation Of Database*

Για να φτιάξουμε την βάση χρησιμοποιήσαμε το περιβάλλον του DB Browser for SQLite.Η βάση δημιουργήθηκε σύμφωνα με το Relational Schema . Δόθηκε ιδιαίτερη προσόχη στην σύνδεση τον κλειδών και εννοιων, και αφου δημιουργήθηκε εξ ολοκλήρου η βάση έγινε έρευνα ως προς το σενάριο ανανέωσης και διαγραφής ενός κύριου κλειδίου(ON UPDATE, ON DELETE).Το περιβάλλον προσφέρει γραφική διαπεφή που κάνει πιο εύκολη την 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 εξής αντιστοίχηση:

* 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 στην 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 την βάση μας με δεδομένα .Tα δεδομένα αυτά βρέθηκαν απο 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 μπορόυν να παρατηρηθούν πιο διεξοδικά και μέσα απο το αρχέιο των CRUD εντολών. Το κομμάτι του DELETE το διαπραγματευτήκαμε φτιάχνοντας βιβλία με λαθος δεδομένα όποτε έπρεπε και να σβηστόυν

Figure 4: Lending a Book Scenario

*2.4.3 Τυπικές Αναζητήσεις*

Αφού ολοκληρώσαμε την ενσωμάτωση δεδομένων στην βάση μας , φτιάξαμε ερωτήματα τυπικών αναζητήσεων παρόμοια με τα πλαίσια του εργαστηρίου του μαθημάτος με σκοπό να αλληλεπιδράσουμε με την βάση δεδομένων και να την επαληθέυσουμε.Στην προσπάθεια μας αυτή δημιουργήθηκαν 38 ερωτήματα απλά και πολύπλοκα, τα οποία αποτελούν test cases για το αν η βάση μας λειτουργεί ικανοποιητικά.Πιο εκτεταμένα ολα τα ερωτήματα μαζί με τις απαντήσεις (queries) βρίσκονται στο αρχείο TestingQueries.sql. Μέσα απο την συγκεκριμένη διαδικάσια καταλάβαμε προβλήματα της βάσης που μας ανάγκασαν να την ξαναδομίσουμε 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 απο την γλώσσα προγραμματισμού python.Χρησιμοποιήσαμε βιβλιοθήκες για GUI (streamlit) , για Data Visualazation (matplotlib) και για την σύνδεση και αλληλεπίδραση με την βάση την sqlite3.Η εφαρμογή αντιπροσωπεύει ένα εσωτερικό σύστημα που θα δίνουμε σε κάθε βιβλιοθήκη του συστήματος μας για να μπορεί να διαχειρήζεται ο εργαζόμενος της ανάγκες της καθημερηνότητας.

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

Ως προς το κομμάτι των ενεργειών για τον τυπικό εργαζόμενο έχουμε δώσει την δυνατότητα να βλέπει την κατάσταση όλων των βιβλιών ή και συγκερκιμένων ανάλογ με φιλτραρίσματα που θα δώσει ο ίδιος, να μπορεί να βάζει ένα καινούργιο βιβλίο στην βιβλιοθήκη που δουλέυει , 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 ΄δωσει ένα πρόστιμο στους πελάτες όπου έχουν αργήσει το βιβλίο.

Απο την άλλη ο διαχειρηστής μπορεί να κάνει αυτά και ακόμα να διαχειρήζεται βιβλιοθήκες μέσα στην βάση , να διαχειρήζεται εργαζόμενους και τμήματα βιβλιοθηκών.Η ανάπτυξη μιας ολοκληρωμένης εφαρμογής στα πλαίσια του εξαμήνου είναι ανέφτική , παρόλο το λίγο χρονικό διάστημα οι πιο βασικές λειτουργίες του δικτύου μας έχουν καληφθεί απο την εφαρμογή.Επειδή ο κώδικας είναι αρκετά εκτενής τον έχουμε προμηθεύσει με αρκετά επεξηγηματικά σχόλια σε όλη την έκταση σου όπου μπορείτε να παρατήσετε και να κατανοήσετε μέσα απο 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 την βάση και της προσθέσαμε δεδομένα για να αξιολογήσουμε πόσο καλά ανταπεξέρχεται σε ενέργειες που πρέπει να γίνουν και γενικότερα στην διαχείρηση των δεδομένων δημιουργήσαμε ερωτήματα τυπικών αναζητήσεων ώστε να δούμε αν τα δεδομένα που επιστρέφονται είναι σωστά.Τρόπος αξιολόγησης είναι η δικία μας άποψη επι της βάσης και της επιστροφής των δεδομένων απο τις τυπικές αναζητήσεις, δηλαδή πρέπει χειροκίνητα να δουμε αν τα δεδομένα που ζητήσαμε είναι τα δεδομένα που επιστραφήκαν , αυτό είναι εφικτό μέσα απο το περιβάλλον της SQLite. Εν τέλη ακόμα ένας τρόπος αξιολόγησης είναι να δούμε το σενάριο του update και delete στα κύρια κλειδία των table μας,δηλάδη τί θα συμβεί στο foreign key όταν το primary key μας αλλάξει ή διαγραφθεί;

Χρησιμοποιήσαμε λοιπόν στον ορίσμό της βάσης την λειτουργία ON UPDATE CASCADE , ON DELETE CASCADE, ON DELETE SET NULL.Καταλήξαμε οτι για το Root Entity όπου είναι η βιβλιοθήκη η διαδικάσια ON DELETE CASCADE έιναι αρκετά επίφοβη γιαυτό και την αγνοήσαμε.Δοκιμάσαμε επίσης ηθηλημένα να βάλουμε λάθος δεδομένα με σκοπό να μας επιστραφεί error.Εν κατακλείδι κύριως τρόπος αξιολόγησης της βάσης δεδομένων για το δίκτυο βιβλιωθηκών είναι τα Test cases απο τις τυπικές αναζητήσεις και τις CRUD εντολές σε συνδυασμό με την δικιά μας αίσθηση πως πρέπει να δουλέυει η βάση.

* 1. Κρητίρια αξιολόγησης

Τα κριτήρια αξιολόγησης διεξάγονται μέσα απο 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 πραγματικά δεδομένα της βάσης, όπου έγινε εφικτό μέσα απο το Data view του SQLite Browser.Μέσω αυτού του κριτηρίου ανακαλύψαμε ότι η βάση μας σε συγκεκριμένες ενέργειες υστερή στην αποδοτικότητα, υπάρχει ίσως μια καλύτερη δομή της βάσης όπου κάποιες ενέργειες θα γινόντουσαν πιο εύκολες.Μια τέτοια ενέργεια είναι η αναζήτηση όλων των βιβλίων που τα έχουν δανείστει.

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

Τελευταίο κριτήριο αξιολόγησης αποτέλεσε η επιτιδευμένη απόπηρα να εισάγουμε λάθος δεδομένα ή δεδομένα εκτός του στατικού μεγέθους όπου έχουμε ορίσει με σκοπό να λάβουμε error .Αυτό μας βοήθησε για ακόμη μια φορά να επαληθέυσουμε την επικοινωνία μεταξύ των table μας και την εγκυρότητα ότι κάτι δεν μπορεί να ορίστει με μεγαλο(επιβλαβές για την βάση) μέγεθος.

1. Δεδομενα

Η εργασία όπως και η αξιολόγηση της βάσης όπως έχει αναφερθεί χρειάστηκε δεδομένα.Κάποια απο αυτά τα δεδομένα τα λάβαμε απο 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 εργαζομένων των βιβλιοθηκών και οι κατηγορίες και τα τμήματα κάθε βιβλιοθήκης.Να διευκρινιστεί ότι τα δεδομένα που φτιάχτηκαν απο εμάς δεν ανταπεξέρχονται σε πραγματικά δεδομένα διότι αγγίζουν ευάισθητα προσωπικά περιεχόμενα (ΑΦΜ, email κλπ).Στην πραγματικότητα αύτα τα δεδομένα θα μπαίνουν encrypted στην βάση ειδικά όταν μιλάμε για δεδομένα κωδικών.

Τα δεδομένα ενσωματόνονται στην 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 απο τις τυπικές αναζητήσεις.Ο δέυτερος ,πιο ρεαλιστικός τρόπος ενσωμάτωσης δεδομένων στην βάση μας, είναι μέσα απο την εφαρμόγη, όπου θα έχουμε καθημερινά interactions με πελάτες , βιβλιοθήκες , εργαζόμενους κλπ.

1. Ολοκληρωση εργασιασ
   1. Κύριες ενέργειες

Η εργασία απο την στιγμή που αποτελείται απο δύο ατόμα θα χρειαστεί μια τεχνολογία αλληλεπίδρασης.Ξεκινήσαμε λοιπόν φτιάχνοντας το δικό μας github repository όπου θα κάνουμε share τον κώδικα μας και θα μπορούμε να δουλεύουμε ασύγχρονα.Εν συνεχεία το phase A (Μικρόκοσμος,ERD,Conceptual Design) το ολοκληρώσαμε απο τις πρώτες εβδομάδες σε συνεργατικό επίπεδο μέσα απο δια ζώσης meetings.Στο phase B ξεκίνησε η παράλληλη δουλεία των μελών της ομάδας με αυτή να χωρίζεται με τέτοιο τρόπο ώστε να μην χαλάει η δουλεία του συνεργαζόμενου.Χωρίσαμε λοιπόν την δουλεία στην δημιουργία της βάσης (όπου έγινε συνεργατηκά) και στην δημιουργία CRUD εντολών (1 άτομο) και δημιουργία τυπικών αναζητήσεων (1 άτομο). Το phase Γ και η έκθεση υλοποιηθήκε απο κοινού με ποσοστίες 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 παρουσιαση |
|  | 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 εισαγωγη της εργασίας.Παρολα αυτά ακριβής διαδικασία για να μπορείτε να τρέξετε τον κώδικα είναι :

1. Κατεβάστε το zip αρχείο του πηγαίου κώδικα απο το github.
2. Καντε 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 απο [εδώ](https://phoenixnap.com/kb/how-to-install-python-3-windows) και βεβαιωθείτε οτι 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 συνδεθέιτε 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 την βάση μέσω της εφαρμογής.Αν θέλετε να ανοίξετε και να τρέξετε τα queries ή να αλληλεπιδράσετε με την 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

REFERENCES

1. Tο GitHub είναι μια αμερικανική εταιρεία που , 210–217. https://doi.org/10.1109/BROADNETS.2004.8
2. Sam Anzaroot and Andrew McCallum. 2013. UMass Citation Field Extraction Dataset. Retrieved May 27, 2019 from <http://www.iesl.cs.umass.edu/data/data-umasscitationfield>
3. Martin A. Fischler and Robert C. Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24, 6 (June 1981), 381–395. https://doi.org/10.1145/358669.358692
4. Chelsea Finn. 2018. Learning to Learn with Gradients. PhD Thesis, EECS Department, University of Berkeley.
5. Jon M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. J. ACM 46, 5 (September 1999), 604–632. https://doi.org/10.1145/324133.324140
6. Matthew Van Gundy, Davide Balzarotti, and Giovanni Vigna. 2007. Catch me, if you can: Evading network signatures with web-based polymorphic worms. In Proceedings of the first USENIX workshop on Offensive Technologies (WOOT ’07) . USENIX Association, Berkley, CA, Article 7, 9 pages.
7. James W. Demmel, Yozo Hida, William Kahan, Xiaoye S. Li, Soni Mukherjee, and Jason Riedy. 2005. Error Bounds from Extra Precise Iterative Refinement. Technical Report No. UCB/CSD-04-1344. University of California, Berkeley.
8. David Harel. 1979. First-Order Dynamic Logic. Lecture Notes in Computer Science, Vol. 68. Springer-Verlag, New York, NY. <https://doi.org/10.1007/3-540-09237-4>
9. Jason Jerald. 2015. The VR Book: Human-Centered Design for Virtual Reality. Association for Computing Machinery and Morgan & Claypool.
10. Prokop, Emily. 2018. The Story Behind. Mango Publishing Group. Florida, USA.
11. R Core Team. 2019. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/
12. Brian K. Reid. 1980. A high-level approach to computer document formatting. In Proceedings of the 7th Annual Symposium on Principles of Programming Languages. ACM, New York, 24–31. <https://doi.org/10.1145/567446.567449>
13. John R. Smith and Shih-Fu Chang. 1997. Visual Seek: a fully automated content-based image query system. In Proceedings of the fourth ACM international conference on Multimedia (MULTIMEDIA ’96). Association for Computing Machinery, New York, NY, USA, 87–98. https://doi.org/10.1145/244130.244151
14. TUG 2017. Institutional members of the LaTeX Users Group. Retrieved May 27, 2017 from <http://wwtug.org/instmem.html>
15. Alper Yilmaz, Omar Javed, and Mubarak Shah. 2006. Object tracking: A survey. ACM Comput. Surv. 38, 4 (December 2006), 13–es. https://doi.org/10.1145/1177352.1177355