



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ
ΑΝΑΦΟΡΑ ΕΞΑΜΗΝΙΑΙΑΣ ΕΡΓΑΣΙΑΣ
Εαρινό εξάμηνο 2022-2023

ΟΜΑΔΑ 13

Ονοματεπώνυμο: Γεώργιος Πιττάκης

Αριθμός Μητρώου: 03120605

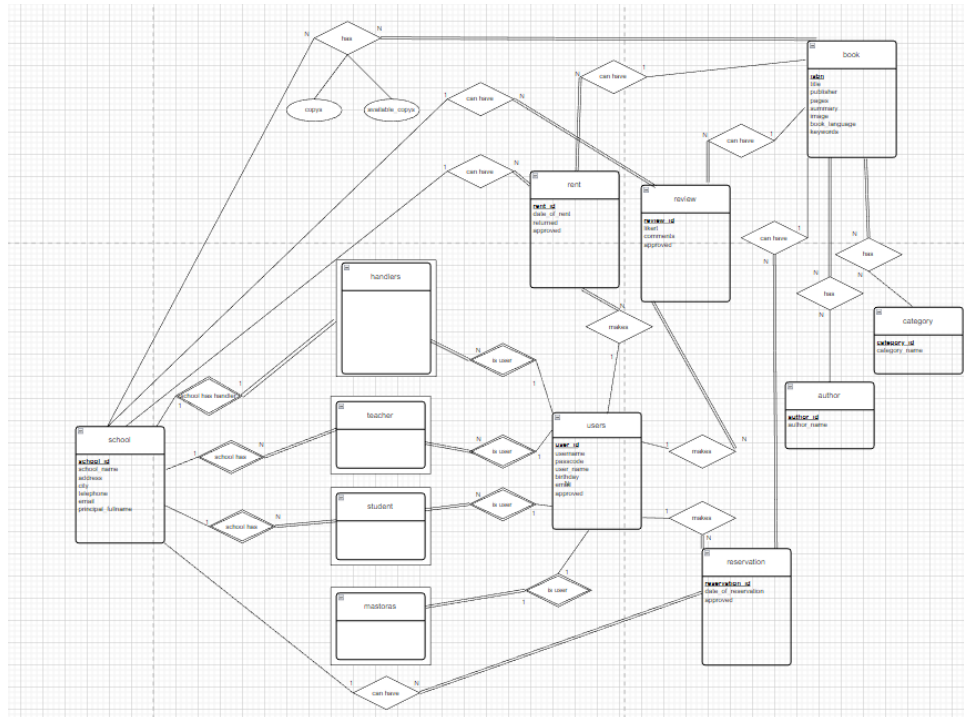
Ονοματεπώνυμο: Χριστόδουλος Στυλιανίδης

Αριθμός Μητρώου: 03120614

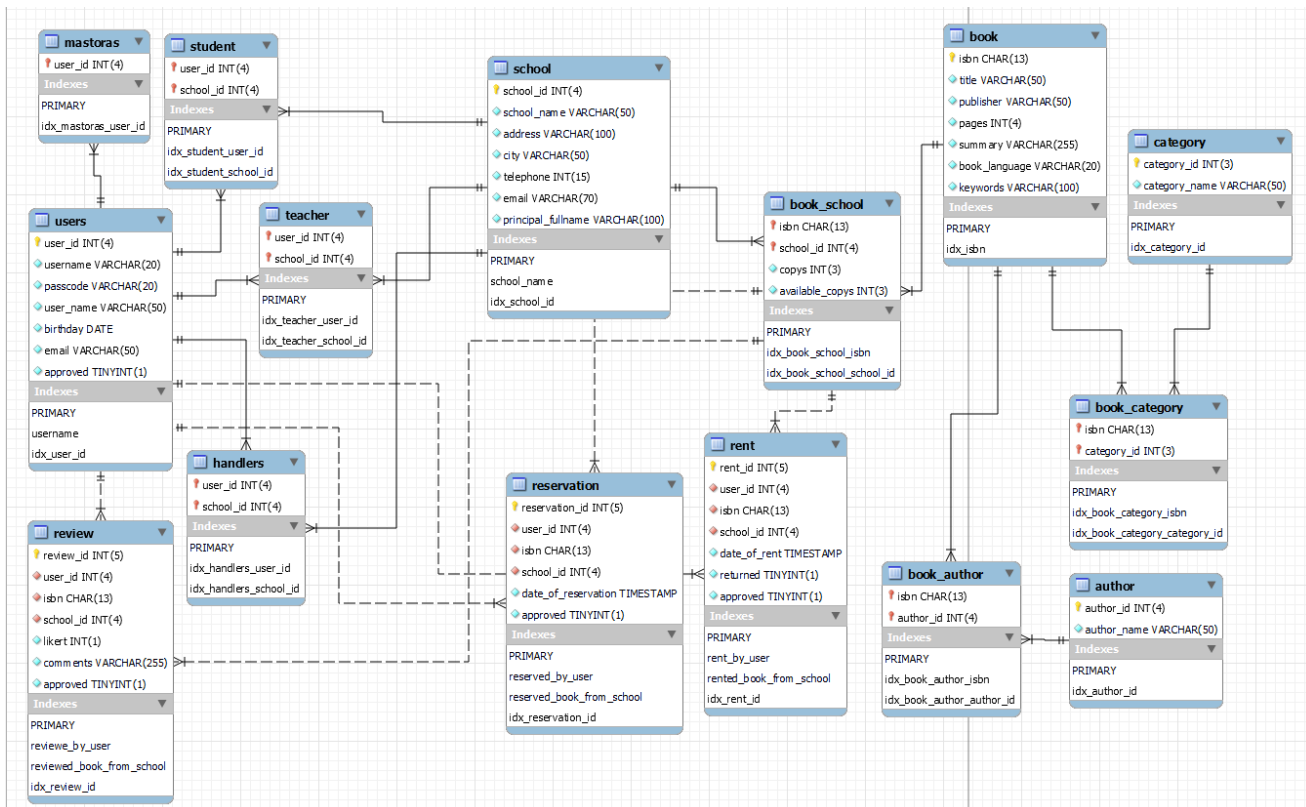
Ονοματεπώνυμο: Ηρακλής Νικολαΐδης

Αριθμός Μητρώου: 03120607

1. ER διάγραμμα



2. Σχισιακό Διάγραμμα – Relational Diagram



Σχολιασμός Σχεσιακού Διαγράμματος

Στην βάση μας περιέχονται πέντε κύριες οντότητες. Οι χρήστες(users), τα σχολεία(school), τα βιβλία(book), οι συγγραφείς(author) και οι κατηγορίες(category). Οι οντότητες που σχετίζονται με τις κύριες κατηγορίες είναι οι book_school, book_author, book_category.

Η κατηγοριοποίηση των χρηστών σε διαχειριστή(mastoras), χειριστές(handlers), καθηγητές(teacher) και μαθητές(student) πραγματοποιήθηκε με τέσσερα επιπλέον tables.

Τέλος οι δανεισμοί(rent), οι κρατήσεις(reservation) και οι αξιολογήσεις(review) έχουν πραγματοποιηθεί με ξεχωριστά tables αφού περιέχουν foreign keys από τα tables users και book_school καθώς και διαφορετικά χαρακτηριστικά για την σχέση, όπως αξιολόγηση σε κλίμακα likert στις αξιολογήσεις, ημερομηνίες δανεισμού/κράτησης στα αντίστοιχα tables κ.α.

Indexing

Για να προσδιορίσουμε τα ευρετήρια σε μια βάση δεδομένων MySQL πρέπει να λάβουμε υπόψη τα εξής:

1. Τα ευρετήρια δημιουργούνται αυτόματα για τις κολώνες που είναι πρωτεύοντα κλειδιά (primary keys) για κάθε πίνακα. Αυτό είναι σημαντικό διότι τα πρωτεύοντα κλειδιά συχνά χρησιμοποιούνται σε ερωτήματα (π.χ. συνθήκες WHERE, JOIN ON). Η γρήγορη πρόσβαση σε αυτά τα πρωτεύοντα κλειδιά επιτυγχάνεται μέσω των ευρετηρίων.
2. Επιπλέον, χρησιμοποιούνται ευρετήρια για τις ξένα κλειδιά (foreign keys) που δεν είναι πρωτεύοντα κλειδιά. Αυτό είναι απαραίτητο για την γρήγορη πρόσβαση κατά την ενημέρωση ή διαγραφή ενός στοιχείου που αναφέρεται από άλλο στοιχείο. Τα ευρετήρια αυτών των ξένων κλειδιών επιτρέπουν την αποδοτική προσπέλαση τους.
3. Για τα σύνθετα πρωτεύοντα κλειδιά που αποτελούνται από πολλές κολώνες, υπάρχει ένα αυτόματο ευρετήριο για το σύνολο των κολώνων αυτών. Ωστόσο, μπορεί επίσης να χρησιμοποιηθεί το ίδιο ευρετήριο και από υποσύνολα αυτών των κολώνων. Για παράδειγμα, αν υπάρχει ένα πρωτεύον κλειδί (a, b, c, d), τότε το ίδιο ευρετήριο μπορεί να χρησιμοποιηθεί και από τις υποσύνολα των κολωνών (a), (a, b), (a, b, c). Ωστόσο, για οποιοδήποτε άλλο υποσύνολο απαιτείται ένα ξεχωριστό ευρετήριο.

Συνολικά, τα ευρετήρια που ορίζονται μπορούν να βρεθούν στο DDL script "parabase.sql" και απεικονίζονται στο σχεσιακό διάγραμμα της βάσης.

Constraints της Βάσης Δεδομένων

Οι περιορισμοί σε αυτήν τη βάση δεδομένων είναι οι εξής:

Περιορισμοί primary key:

Ο περιορισμός PRIMARY KEY εφαρμόζεται στους παρακάτω πίνακες:

- Ο πίνακας "school" στη στήλη "school_id".
- Ο πίνακας "author" στη στήλη "author_id".
- Ο πίνακας "category" στη στήλη "category_id".
- Ο πίνακας "book" στη στήλη "isbn".
- Ο πίνακας "book_author" στον συνδυασμό των στηλών "isbn" και "author_id".
- Ο πίνακας "book_school" στον συνδυασμό των στηλών "isbn" και "school_id".
- Ο πίνακας "book_category" στον συνδυασμό των στηλών "isbn" και "category_id".
- Ο πίνακας "users" στη στήλη "user_id".
- Ο πίνακας "student" στον συνδυασμό των στηλών "user_id" και "school_id".
- Ο πίνακας "teacher" στον συνδυασμό των στηλών "user_id" και "school_id".
- Ο πίνακας "handlers" στον συνδυασμό των στηλών "user_id" και "school_id".
- Ο πίνακας "mastoras" στη στήλη "user_id".
- Ο πίνακας "rent" στη στήλη "rent_id".
- Ο πίνακας "reservation" στη στήλη "reservation_id".
- Ο πίνακας "review" στη στήλη "review_id".

Περιορισμοί foreign key:

Περιορισμοί FOREIGN KEY εφαρμόζονται για τη δημιουργία συσχετίσεων μεταξύ πινάκων. Οι παρακάτω πίνακες έχουν περιορισμούς foreign key:

- Ο πίνακας "book_author" έχει το πεδίο "isbn" να αναφέρεται στο πεδίο "isbn" του πίνακα "book" και το πεδίο "author_id" να αναφέρεται στο πεδίο "author_id" του πίνακα "author".
- Ο πίνακας "book_school" έχει το πεδίο "isbn" να αναφέρεται στο πεδίο "isbn" του πίνακα "book" και το πεδίο "school_id" να αναφέρεται στο πεδίο "school_id" του πίνακα "school".
- Ο πίνακας "book_category" έχει το πεδίο "isbn" να αναφέρεται στο πεδίο "isbn" του πίνακα "book" και το πεδίο "category_id" να αναφέρεται στο πεδίο "category_id" του πίνακα "category".
- Ο πίνακας "student" έχει το πεδίο "user_id" να αναφέρεται στο πεδίο "user_id" του πίνακα "users" και το πεδίο "school_id" να αναφέρεται στο πεδίο "school_id" του πίνακα "school".
- Ο πίνακας "teacher" έχει το πεδίο "user_id" να αναφέρεται στο πεδίο "user_id" του πίνακα "users" και το πεδίο "school_id" να αναφέρεται στο πεδίο "school_id" του πίνακα "school".
- Ο πίνακας "handlers" έχει το πεδίο "user_id" να αναφέρεται στο πεδίο "user_id" του πίνακα "users" και το πεδίο "school_id" να αναφέρεται στο πεδίο "school_id" του πίνακα "school".

- Ο πίνακας "rent" έχει το πεδίο "user_id" να αναφέρεται στο πεδίο "user_id" του πίνακα "users" και τα πεδία "isbn" και "school_id" να αναφέρονται στα πεδία "isbn" και "school_id" του πίνακα "book_school".
- Ο πίνακας "reservation" έχει το πεδίο "user_id" να αναφέρεται στο πεδίο "user_id" του πίνακα "users" και τα πεδία "isbn" και "school_id" να αναφέρονται στα πεδία "isbn" και "school_id" του πίνακα "book_school".
- Ο πίνακας "review" έχει το πεδίο "user_id" να αναφέρεται στο πεδίο "user_id" του πίνακα "users" και τα πεδία "isbn" και "school_id" να αναφέρονται στα πεδία "isbn" και "school_id" του πίνακα "book_school".

Μοναδικός Περιορισμός:

Ο μοναδικός περιορισμός (UNIQUE constraint) εφαρμόζεται στη στήλη "school_name" του πίνακα "school" για να εξασφαλιστεί η μοναδικότητα των ονομάτων σχολείων και στη στήλη username του πίνακα "users" για να εξασφαλιστεί η μοναδικότητα των username.

Περιορισμοί Ελέγχου:

Οι περιορισμοί ΕΛΕΓΧΟΥ (CHECK constraints) εφαρμόζονται στον πίνακα "book_school" για την επιβολή των ακόλουθων συνθηκών:

- available_copys <= copys: Τα διαθέσιμα αντίτυπα πρέπει να είναι μικρότερα ή ίσα με τον συνολικό αριθμό αντιτύπων.
- copys > 0: Ο συνολικός αριθμός αντιτύπων πρέπει να είναι μεγαλύτερος του μηδενός.
- available_copys >= 0: Τα διαθέσιμα αντίτυπα πρέπει να είναι μεγαλύτερα ή ίσα με το μηδέν.

Οι ρήτρες ON DELETE CASCADE ON UPDATE CASCADE στα ορισμένα foreign keys υποδηλώνουν ότι όταν διαγράφεται ή ενημερώνεται μια αναφερόμενη γραμμή, οι αντίστοιχες γραμμές στον αναφέροντα πίνακα θα διαγραφούν ή ενημερωθούν αντίστοιχα.

2.1 DDL και DML scripts

DDL scripts :

Στην βάση δεδομένων η οποία βρίσκεται στον φάκελο papabase_sql στο repository στο github υπάρχει ένα αρχείο DDL script το papabase.sql.

```
1 DROP DATABASE IF EXISTS library_project;
2 CREATE USER IF NOT EXISTS 'papamaster'@'localhost' IDENTIFIED BY 'password';
3 CREATE DATABASE library_project;
4 GRANT ALL PRIVILEGES ON library_project.* TO 'papamaster'@'localhost';
5
6 USE library_project;
7
8 CREATE TABLE school(
9     school_id integer(4) NOT NULL auto_increment,
10    school_name varchar(50) NOT NULL UNIQUE,
11    address varchar(100) NOT NULL,
12    city varchar(50) NOT NULL,
13    telephone integer(15) NOT NULL,
14    email varchar(70) NOT NULL,
15    principal_fullname varchar(50) NOT NULL,
16    PRIMARY KEY(school_id)
17 );
18
19 CREATE TABLE reservation(
20     reservation_id integer(5) NOT NULL auto_increment,
21     user_id integer(4) NOT NULL,
22     isbn char(13) NOT NULL,
23     school_id integer(4) NOT NULL,
24     date_of_reservation timestamp NOT NULL,
25     approved boolean NOT NULL DEFAULT 0,
26     CONSTRAINT reserved_by_user FOREIGN KEY (user_id)
27     REFERENCES users(user_id) ON DELETE CASCADE ON UPDATE CASCADE,
28     CONSTRAINT reserved_book FOREIGN KEY (isbn,school_id)
29     REFERENCES book_school(isbn,school_id) ON DELETE CASCADE ON UPDATE CASCADE,
30     PRIMARY KEY (reservation_id)
31 );
32
```

Σε αυτό το script η πρώτη γραμμή διαγράφει ολόκληρη την βάση εάν υπάρχει και στη συνέχεια δημιουργούμε τη βάση και τα tables όπως φαίνεται πιο πάνω. Επίσης δημιουργούνται όλες οι primary key εξαρτήσεις και τα check constraints της βάσης.

DML scripts :

Στην βάση δεδομένων η οποία βρίσκεται στον φάκελο papabase_sql στο repository στο github υπάρχει ένα αρχείο DML script το papa_generator.sql το οποίο περιλαμβάνει τα αρχικά δεδομένα που εισάγουμε στη βάση . Στην κατηγορία αυτή συμπεριλαμβάνονται επίσης τα queries που ζητούνται στην εκφώνηση και οι όψεις της βάσης.

3 Queries

3.1 Διαχειριστής

3.1.1 Λίστα με συνολικό αριθμό δανεισμών ανά σχολείο (Κριτήρια αναζήτησης: έτος (rent_year), ημερολογιακός μήνας (rent_month))

```
SELECT s.school_id, s.school_name, COUNT(*) AS total_loans
FROM rent AS r
JOIN school AS s ON r.school_id = s.school_id
WHERE YEAR(r.date_of_rent) = rent_year AND MONTH(r.date_of_rent) = rent_month
GROUP BY r.school_id
```

3.1.2 Για δεδομένη κατηγορία βιβλίων που επιλέγει ο χρήστης, ποιοι συγγραφείς ανήκουν σε αυτήν και ποιοι εκπαιδευτικοί έχουν δανειστεί βιβλία αυτής της κατηγορίας το τελευταίο έτος.

```
SELECT DISTINCT u.user_id, u.user_name
FROM users u
JOIN teacher t ON u.user_id = t.user_id
JOIN rent r ON u.user_id = r.user_id
JOIN book_school bs ON r.isbn = bs.isbn AND r.school_id = bs.school_id
JOIN book_category bc ON bs.isbn = bc.isbn
JOIN category c ON bc.category_id = c.category_id
WHERE c.category_id = categoryid
AND r.date_of_rent >= DATE_SUB(NOW(), INTERVAL 1 YEAR)
GROUP BY u.user_id;
```

```
SELECT DISTINCT a.author_name, a.author_id
FROM author a
JOIN book_author ba ON a.author_id = ba.author_id
JOIN book_category bc ON ba.isbn = bc.isbn
JOIN category c ON bc.category_id = c.category_id
WHERE c.category_id = categoryid
GROUP BY a.author_id
```

3.1.3 Βρείτε τους νέους εκπαιδευτικούς (ηλικία < 40 ετών) που έχουν δανειστεί τα περισσότερα βιβλία και τον αριθμό των βιβλίων.

```
SELECT u.user_id, u.user_name, u.birthday, COUNT(*) AS book_count
FROM users u
    JOIN teacher t ON u.user_id = t.user_id
    JOIN rent r ON u.user_id = r.user_id
    JOIN book_school bs ON r.isbn = bs.isbn AND r.school_id = bs.school_id
WHERE TIMESTAMPDIFF(YEAR, u.birthday, CURDATE()) < 40
GROUP BY u.user_id, u.user_name, u.birthday
HAVING COUNT(*) = (
    SELECT COUNT(*)
    FROM users uu
        JOIN teacher tt ON uu.user_id = tt.user_id
        JOIN rent rr ON uu.user_id = rr.user_id
        JOIN book_school bb ON rr.isbn = bb.isbn AND rr.school_id = bb.school_id
    WHERE TIMESTAMPDIFF(YEAR, uu.birthday, CURDATE()) < 40
    GROUP BY uu.user_id
    ORDER BY COUNT(*) DESC LIMIT 1)
```

3.1.4 Βρείτε τους συγγραφείς των οποίων κανένα βιβλίο δεν έχει τύχει δανεισμού

```
SELECT a.author_id, a.author_name
FROM author a
WHERE NOT EXISTS (
    SELECT 1
    FROM book_author ba
    INNER JOIN book b ON ba.isbn = b.isbn
    INNER JOIN rent r ON b.isbn = r.isbn
    WHERE ba.author_id = a.author_id )
```

3.1.5 Ποιοι χειριστές έχουν δανείσει τον ίδιο αριθμό βιβλίων σε διάστημα ενός έτους με περισσότερους από 20 δανεισμούς;

```
SELECT h.user_id, u.user_name, h.school_id, COUNT(DISTINCT CASE WHEN r.approved = 1 THEN r.rent_id END) AS rented_books
FROM handlers h
    JOIN rent r ON h.school_id = r.school_id
    JOIN users u ON h.user_id = u.user_id
WHERE r.date_of_rent >= DATE_SUB(CURDATE(), INTERVAL 1 YEAR)
GROUP BY h.user_id, u.user_name, h.school_id
HAVING rented_books > 20
AND rented_books IN (
    SELECT COUNT(DISTINCT CASE WHEN r2.approved = 1 THEN r2.rent_id END)
    FROM rent r2
    WHERE r2.date_of_rent >= DATE_SUB(CURDATE(), INTERVAL 1 YEAR)
    GROUP BY r2.school_id
    HAVING COUNT(DISTINCT CASE WHEN r2.approved = 1 THEN r2.rent_id END) > 20
    AND r2.school_id <> h.school_id )
```


- 3.1.6 Πολλά βιβλία καλύπτουν περισσότερες από μια κατηγορίες. Ανάμεσα σε ζεύγη πεδίων (π.χ. ιστορία και ποίηση) που είναι κοινά στα βιβλία, βρείτε τα 3 κορυφαία (top-3) ζεύγη που εμφανίστηκαν σε δανεισμούς.

```
SELECT c1.category_name AS category1, c2.category_name AS category2
FROM (
  SELECT bc1.category_id AS category_id1, bc2.category_id AS category_id2, COUNT(*) AS combination_count
  FROM book_category bc1
    JOIN book_category bc2 ON bc1.isbn = bc2.isbn AND bc1.category_id < bc2.category_id
    JOIN book b ON bc1.isbn = b.isbn
    JOIN rent r ON b.isbn = r.isbn
  GROUP BY bc1.category_id, bc2.category_id
) AS combinations
JOIN category c1 ON combinations.category_id1 = c1.category_id
JOIN category c2 ON combinations.category_id2 = c2.category_id
GROUP BY category1, category2
ORDER BY combination_count DESC LIMIT 3
```

- 3.1.7 Βρείτε όλους τους συγγραφείς που έχουν γράψει τουλάχιστον 5 βιβλία λιγότερα από τον συγγραφέα με τα περισσότερα βιβλία.

```
SELECT a1.author_id, a1.author_name
FROM author a1
WHERE (
  SELECT COUNT(*)
  FROM book_author ba1
  WHERE ba1.author_id = a1.author_id
) <= (
  SELECT COUNT(*)
  FROM book_author ba2
  GROUP BY ba2.author_id
  ORDER BY COUNT(*) DESC
  LIMIT 1) - 5
```

3.2 Χειριστής

3.2.1 Παρουσίαση όλων των βιβλίων κατά Τίτλο, Συγγραφέα (Κριτήρια αναζήτησης: τίτλος/ κατηγορία/ συγγραφέας/ αντίτυπα).

```
SELECT b.title, b.isbn, bs.available_copys, GROUP_CONCAT(a.author_name) AS authors
FROM book b
JOIN book_school bs ON b.isbn = bs.isbn
JOIN book_category bc ON b.isbn = bc.isbn
JOIN category c ON bc.category_id = c.category_id
JOIN book_author ba ON b.isbn = ba.isbn
JOIN author a ON ba.author_id = a.author_id
JOIN handlers h ON bs.school_id = h.school_id
WHERE h.user_id = ?

AND c.category_name = ? #Afta prosthetontai
AND b.title = ? #analogia me ta kritiria
AND a.author_name = ? #anazitisis pou vazi o
AND bs.copys = ? #xiristis

GROUP BY b.title, b.isbn, bs.available_copys
```

3.2.2 Εύρεση όλων των δανειζόμενων που έχουν στην κατοχή τους τουλάχιστον ένα βιβλίο και έχουν καθυστερήσει την επιστροφή του. (Κριτήρια αναζήτησης: Όνομα, Επώνυμο, Ημέρες Καθυστερήσης).

```
SELECT u.user_id, u.username, u.user_name, (SELECT title FROM book WHERE isbn = r.isbn) AS title, r.title, r.school_id, DATEDIFF(NOW(), r.date_of_rent) - 7 AS days_passed
FROM users u
INNER JOIN rent r ON u.user_id = r.user_id
INNER JOIN handlers h ON h.school_id = r.school_id
WHERE h.user_id = userid
AND r.approved = 1
AND r.returned = 0
AND DATEDIFF(NOW(), r.date_of_rent) > 7
AND u.user_name = namee
AND DATEDIFF(NOW(), r.date_of_rent) - 7 = dayz
```

3.2.3 Μέσος Όρος Αξιολογήσεων ανά δανειζόμενο και κατηγορία (Κριτήρια αναζήτησης: χρήστης/ κατηγορία)

```
SELECT u.username, c.category_name, AVG(r.likert) AS average_likert
FROM review r
JOIN users u ON r.user_id = u.user_id
JOIN book_category bc ON r.isbn = bc.isbn
JOIN category c ON bc.category_id = c.category_id
JOIN book_school bs ON r.isbn = bs.isbn
JOIN handlers h ON bs.school_id = h.school_id
WHERE r.school_id = (SELECT school_id FROM handlers WHERE user_id = ?)
AND h.user_id = ?
GROUP BY u.username, c.category_name
```

3.3 Χρήστης

3.3.1 Όλα τα βιβλία που έχουν καταχωριστεί (Κριτήρια αναζήτησης: τίτλος/ κατηγορία/ συγγραφέας), δυνατότητα επιλογής βιβλίου και δημιουργία αιτήματος κράτησης.

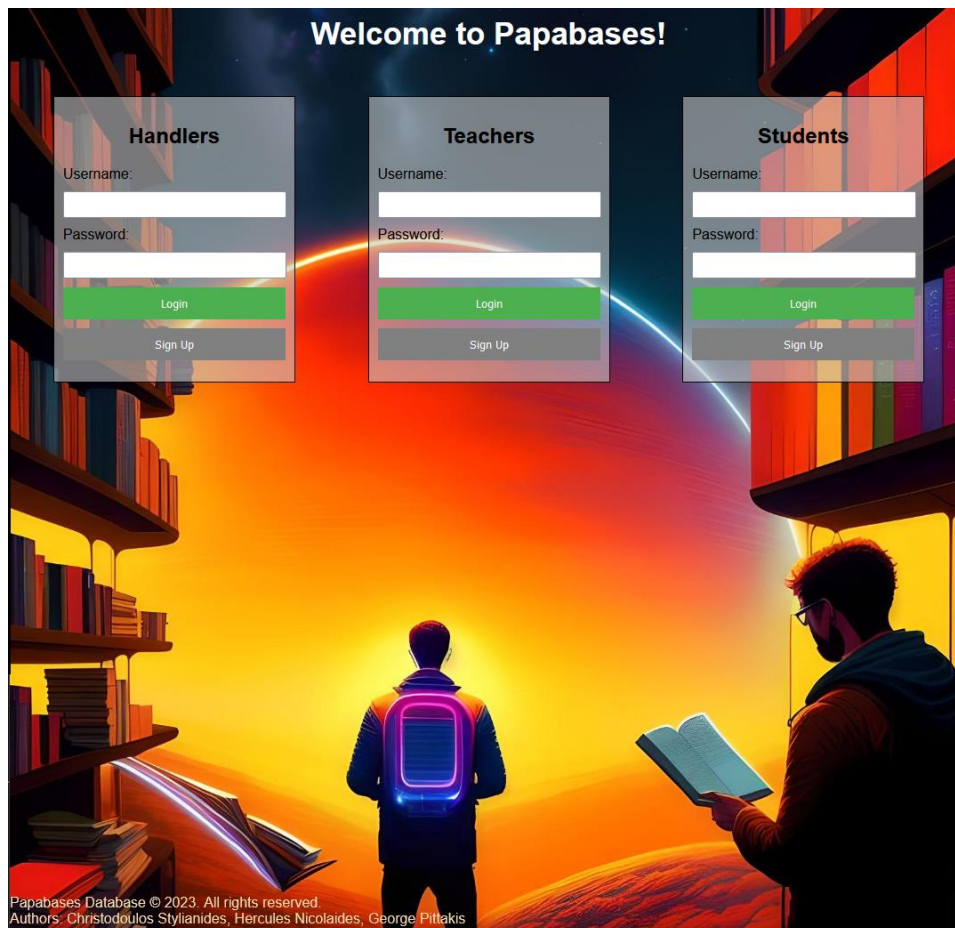
```
SELECT bs.isbn, b.title, GROUP_CONCAT(DISTINCT a.author_name) AS author_names, GROUP_CONCAT(DISTINCT c.category_name) AS category_names,
       bs.school_id, b.publisher, b.pages, b.summary, b.book_language, b.keywords, bs.available_copys
FROM book_school bs
JOIN book b ON bs.isbn = b.isbn
JOIN book_author ba ON b.isbn = ba.isbn
JOIN author a ON ba.author_id = a.author_id
JOIN book_category bc ON b.isbn = bc.isbn
JOIN category c ON bc.category_id = c.category_id
WHERE bs.school_id = schoolid
      AND (c.category_name = 'categoryname'           #analoga me ta ksritiria
      AND b.title = 'title'                           #fevgi kai kapia sinthiki
      AND a.author_name = 'authorname' )              #meta to WHERE
GROUP BY bs.isbn, b.title, bs.school_id, b.publisher, b.pages, b.summary, b.book_language, b.keywords, bs.available_copys
```

3.3.2 Λίστα όλων των βιβλίων που έχει δανειστεί ο συγκεκριμένος χρήστης.

```
SELECT * FROM rent AS r
JOIN book_school AS bs ON r.isbn = bs.isbn AND r.school_id = bs.school_id
JOIN book AS b ON bs.isbn = b.isbn
WHERE r.user_id = userid
```

4 Πλοήγηση στο User Interface

Μέσο της ιστοσελίδας ο χρήστης μπορεί να συνδεθεί στον υπάρχον λογαριασμό του ή να κάνει εγγραφή στο σύστημα αν δεν έχει ήδη λογαριασμό ανάλογα με την ιδιότητα του σε ένα από τα τρία πεδία.



Student Signup

Username:

Password:

Name and Surname:

Birth day:

dd/mm/yyyy

Email:

School:

Select a school

Sign Up

Αφού γίνει η εγγραφή και εγκριθεί ο λογαριασμός, ο χρήστης μπορεί με την σύνδεση στον λογαριασμό του να δει τα στοιχεία του και να κάνει χρήση των λειτουργιών της ιστοσελίδας.

[Logout](#)

My Profile

User Type: Handler
School: Lukeio Paralimniou
User ID: 4
Username: pgeorgiou
Password: *****
Name-Surname: Panayiotis Georgiou
Email: pgeorgiou@gmail.com
Birthday: 1976-02-07

Approve Students 0
[Show](#)

Approve Teachers 0
[Show](#)

Approve Reviews 4
[Show](#)

Approve Rents 0
[Show](#)

Approve Reservations 0
[Show](#)

Book Returns 0
[Show](#)

The list of all books
[Show](#)

Average likert
[Show](#)

Delete or Deactivate Users
[Show](#)

Users with a delayed return
[Show](#)

[Logout](#)

My Profile

User Type: Student
School: Lukeio Paralimniou
User ID: 49
Username: GiorgosP77
Password: *****
Name-Surname: Giorgos Pittakis
Email: gpittakis@gmail.com
Birthday: 2001-10-29

The list of books in my school
[Show](#)

My Rents
[Show](#)

My Reservations
[Show](#)

Rent a book
[Rent](#)

Reserve a book
[Reserve](#)

Review of a book
[Review](#)

5 Οδηγίες Εγκατάστασης

Το repository της βάσης δεδομένων βρίσκεται στο github :

<https://github.com/ChrisOPellouis/Papabases>

- Αρχικά εγκαθιστούμε τα git bash, node js.
- Για την εγκατάσταση της εφαρμογής θα πρέπει να γίνει clone το repository της εφαρμογής μέσω του terminal με την εντολή :

```
git clone https://github.com/ChrisOPellouis/Papabases.git
```

- Έπειτα αφού κατεβάσουμε το mySQL Workbench και mySQL Server τρέχουμε το αρχείο papabase.sql και μετά το αρχείο papa_generator.sql.
- Στη συνέχεια γράφουμε στο terminal τις εντολές:

```
pip install -r requirements.txt  
npm i express  
npm install -g npm@9.6.7  
python3
```

- Σε ξεχωριστά terminal, πλοηγούμαστε στο φάκελο Papabases γράφουμε τις εντολές:
 1. cd backend
 2. node server.js
- Τέλος ανοίγουμε έναν browser και βάζουμε την διεύθυνση :

<http://localhost:3000>

