



## **ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ**

**ΕΞΑΜΗΝΙΑΙΑ ΕΡΓΑΣΙΑ**

**Ακαδημαϊκό Έτος 2022-2023**

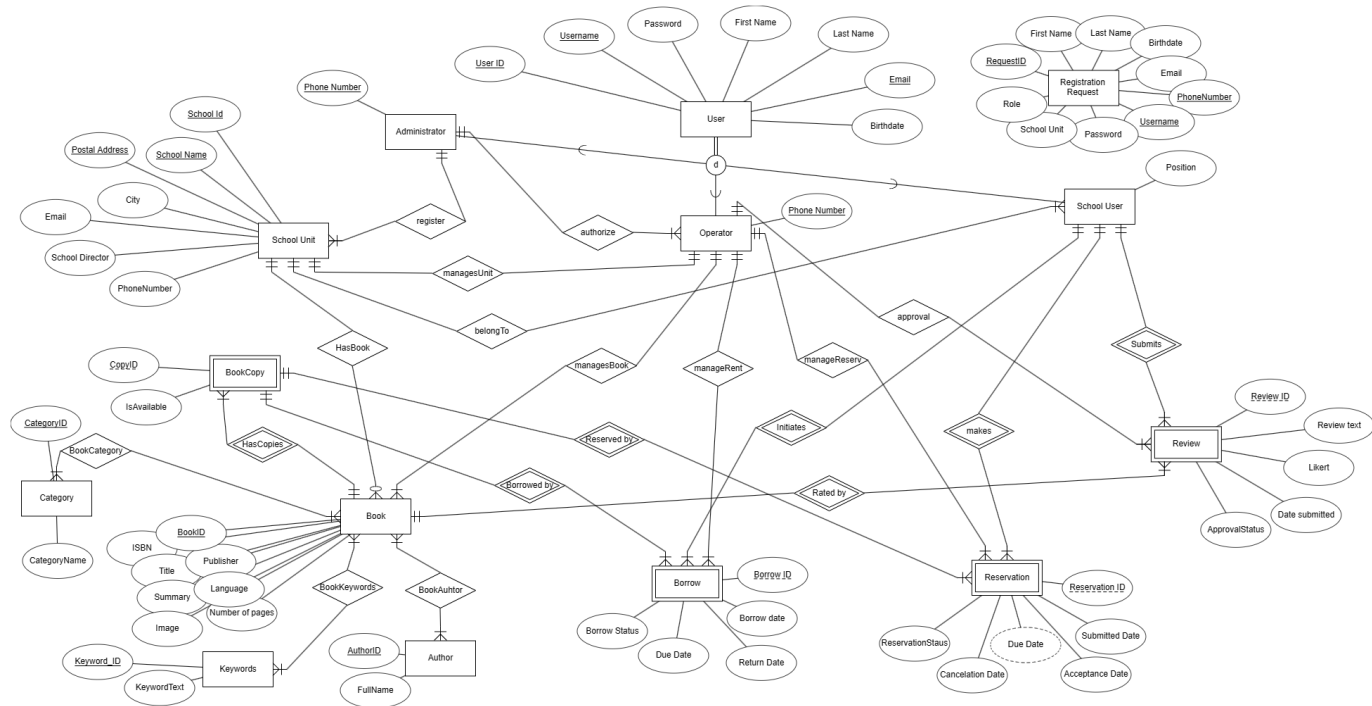
**Ομάδα Project 23**

**Μερεκούλιας Χαράλαμπος | ge19082**

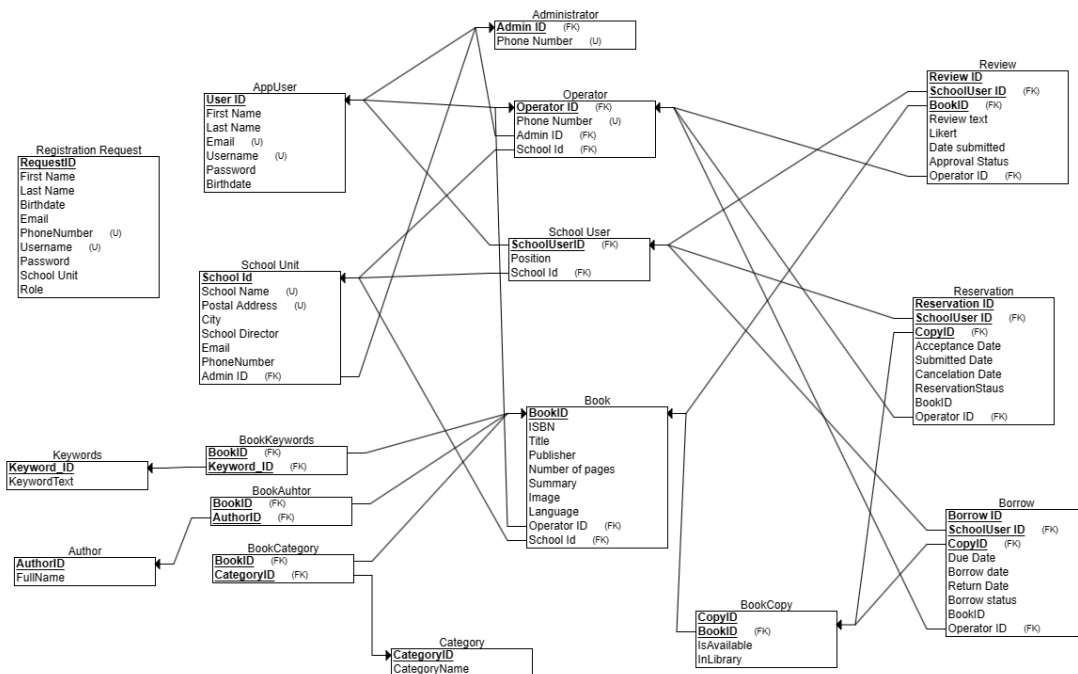
**Νότσικα Τηλέμαχος | ge19050**

**Πέτσα Χριστίνα | ge19016**

## E-R Διάγραμμα



## Σχεσιακό Διάγραμμα



## Σχολιασμός E-R & Σχεσιακού Διαγραμμάτων

Στο E-R διάγραμμα περιέχονται 5 κύριες οντότητες: ο Administrator, ο Operator, ο School User(αντιπροσωπεύει μαθητές και καθηγητές), το School Unit και το Book.

- Οι οντότητες **Administrator**, ο **School User**, ο **School User** ενώνονται με σχέση total specialization με την οντότητα **User** (ή AppUser στο σχεσιακό), η οποία δημιουργήθηκε με σκοπό να κρατά τα κοινά attributes των οντοτήτων αυτών και να επιτρέπει ο κάθε χρήστης να μπορεί να έχει ξεχωριστό ρόλο και attributes μοναδικά στο specialization του. Επίσης, έχει νόημα να υπάρχει μία γενική οντότητα που αντιπροσωπεύει όλους τους χρήστες.
- Επιπλέον, υπάρχει η οντότητα **Registration Request**, που κρατά τα requests που γίνονται και τα οποία πρέπει να μπορούν να διαχειρίζονται ο Administrator και ο Operator. Η οντότητα αυτή υπάρχει και για λόγους ασφάλειας για να μην έχει κάποιος πρόσβαση σε δεδομένα της βάση χωρίς να είναι χρήστης και συνεπώς δεν συνδέεται με σχέση με την υπόλοιπη βάση.
- Η οντότητα **School Unit** αντιπροσωπεύει τις σχολικές μονάδες και κατά επέκταση τις σχολικές βιβλιοθήκες του συστήματος βιβλιοθηκών.
- Η οντότητα **Book** αντιπροσωπεύει όλα τα βιβλία που υπάρχουν σε όλες τις βιβλιοθήκες σε επίπεδο βιβλιοθήκης, δηλαδή αν ένα βιβλίο υπάρχει τουλάχιστον μία φορά σε μία βιβλιοθήκη θα υπάρχει μία φορά ως tuple στην οντότητα Book και θα αντιστοιχεί στην συγκεκριμένη μόνο βιβλιοθήκη-σχολική μονάδα όπου υπάρχει. Αν το ίδιο υπάρχει και σε άλλη βιβλιοθήκη θα υπάρχει άλλο tuple στο Book που το αντιπροσωπεύει.
- Τα copies των βιβλίων αντιπροσωπεύονται από την weak entity **BookCopy**, που σχετίζεται άμεσα με την οντότητα Book, και έτσι για κάθε φυσικό αντίγραφο ενός βιβλίου που υπάρχει σε μία σχολική βιβλιοθήκη υπάρχει και ένα στην weak entity BookCopy.
- Υπάρχουν 2 weak entities που σχετίζονται με το BookCopy και το Student, αυτά είναι οι οντότητες: **Borrow** και **Reservation**, που αντιπροσωπεύουν αντίστοιχα τους δανεισμούς και τις κρατήσεις φυσικών αντιγράφων βιβλίων από τους μαθητές.
- Επίσης υπάρχει το weak entity **Review**, που αντιπροσωπεύει τις αξιολογήσεις όλων των μαθητών για ένα βιβλίο μίας σχολικής μονάδας και για αυτό σχετίζεται με Book και όχι με το BookCopy, αφού η γνώση ποιανού copy είναι η κάθε αξιολόγηση δεν είναι χρήσιμη.
- Τέλος υπάρχουν οντότητες που αντιπροσωπεύουν στην ουσία attributes που μπορεί, όμως να έχουν πολλαπλές τιμές (multivariable attributes). Αυτά είναι τα εξής: **Author**, **Keywords**, **Category**.

Το Σχεσιακό Διάγραμμα υλοποιείται αντίστοιχα και με βάση το E-R διάγραμμα, αναπαριστά όλες τις οντότητες μέσα στην βάση με τα attributes τους.

## Indexing Βάσης Δεδομένων

Ορίζονται ευρετήρια στα attributes στα οποία θέλουμε να έχουμε εύκολη και γρήγορη πρόσβαση, καθώς η χρήση τους είναι συχνή. Αυτά είναι τα εξής:

- ✓ (Username, Password) για κάθε χρήστη
- ✓ (SchoolID, SchoolUserID, Position) για κάθε χρήστη
- ✓ (BookID, Title, SchoolID) για κάθε βιβλίο
- ✓ (CategoryName) για κάθε κατηγορία
- ✓ (SchoolUserID, Borrow\_Date, BorrowID, BookID) για κάθε δανεισμό
- ✓ (SchoolUserID, BookID, OperatorID) για κάθε αξιολόγηση

Στο 1.2. παρουσιάζεται και ο SQL κώδικας για την δημιουργία των indices.

### 1.2.

#### DDL Script

Στην δημιουργία των πινάκων του Σχεσιακού Διαγράμματος στην SQL, χρησιμοποιήσαμε το NOT NULL για την αρχικοποίηση κάθε στοιχείου, όταν αυτό κρινόταν απαραίτητο, όπως στα foreign keys, που δεν πρέπει να είναι NULL, για να συμφωνούν με τα primary keys των σχέσεων στις οποίες ανήκουν.

Στον φάκελο DDL\_SCRIPT.sql, στην βάση δεδομένων, υπάρχει ο απαραίτητος κώδικας SQL για την δημιουργία της βάσης μας και των κατάλληλων indices, όπως παρουσιάζεται παρακάτω.

#### Query για την δημιουργία της βάσης

```
create database myschool_library;  
use myschool_library;
```

#### Query για την δημιουργία του πίνακα AppUser

```
CREATE TABLE IF NOT EXISTS AppUser(
  UserID INT not null AUTO_INCREMENT,
  FirstName VARCHAR(45) not null,
  LastName VARCHAR(45) not null,
  BirthDate DATE not null,
  Email VARCHAR(80) UNIQUE not null,
  Username VARCHAR(80) UNIQUE not null,
  Password VARCHAR(100) not null,
  PRIMARY KEY (UserID)
);
```

Το UserID είναι το Primary Key, γιατί είναι μοναδικό attribute για κάθε χρήστη.

### **Query για την δημιουργία του πίνακα Administrator**

```
CREATE TABLE IF NOT EXISTS Administrator(
  AdminID INT not null,
  PhoneNumber VARCHAR(45) UNIQUE not null,
  PRIMARY KEY (AdminID),
  FOREIGN KEY (AdminID) REFERENCES AppUser(UserID) on DELETE RESTRICT on UPDATE CASCADE
);
```

Η στήλη AdminID στον πίνακα Administrator λειτουργεί τόσο ως πρωτεύον όσο και ως ξένο κλειδί του πίνακα. Ως πρωτεύον κλειδί, προσδιορίζει μοναδικά κάθε γραμμή του πίνακα. Ως ξένο κλειδί, παραπέμπει στη στήλη UserID του πίνακα AppUser, πράγμα που σημαίνει ότι κάθε διαχειριστής πρέπει να είναι και χρήστης. Ο κώδικας καθορίζει επίσης τις ενέργειες που πρέπει να γίνονται όταν τα δεδομένα στα οποία γίνεται αναφορά τροποποιούνται ή διαγράφονται. Η ρήτρα on DELETE RESTRICT σημαίνει ότι αν διαγραφεί ένας χρήστης από τον πίνακα AppUser, ο αντίστοιχος διαχειριστής δεν μπορεί να διαγραφεί από τον πίνακα Administrator. Η ρήτρα on UPDATE CASCADE σημαίνει ότι εάν ενημερωθεί ένα αναγνωριστικό χρήστη στον πίνακα AppUser, το αντίστοιχο αναγνωριστικό διαχειριστή θα ενημερωθεί και στον πίνακα Administrator. Με αυτόν τον τρόπο, ο κώδικας διατηρεί τη συνοχή και την ακεραιότητα των δεδομένων στους δύο πίνακες.

### **Query για την δημιουργία του πίνακα School Unit**

```

CREATE TABLE IF NOT EXISTS School_Unit(
SchoolID INT not null AUTO_INCREMENT,
School_Name VARCHAR(72) UNIQUE not null,
Postal_Address VARCHAR(72) UNIQUE not null,
City VARCHAR(45) not null,
PhoneNumber VARCHAR(45) UNIQUE not null,
Email VARCHAR(80) UNIQUE not null,
School_Director VARCHAR(45) not null,
AdminID INT not null,
PRIMARY KEY (SchoolID),
FOREIGN KEY (AdminID) REFERENCES Administrator(AdminID) on DELETE RESTRICT on UPDATE CASCADE
);

```

SchoolID: Πρόκειται για το πρωτεύον κλειδί του πίνακα που προσδιορίζει μοναδικά κάθε σχολείο. Είναι ένας ακέραιος αριθμός που αυξάνεται αυτόματα από το σύστημα της βάσης δεδομένων.

AdminID: Πρόκειται για ένα ξένο κλειδί που παραπέμπει στη στήλη AdminID του πίνακα Administrator. Αυτό σημαίνει ότι κάθε σχολείο πρέπει να έχει έναν διαχειριστή που είναι επίσης χρήστης στον πίνακα AppUser. Ο κώδικας καθορίζει επίσης τις ενέργειες που πρέπει να γίνονται όταν τα δεδομένα στα οποία γίνεται αναφορά τροποποιούνται ή διαγράφονται. Η ρήτρα on DELETE RESTRICT σημαίνει ότι εάν ένας διαχειριστής διαγραφεί από τον πίνακα Administrator, το αντίστοιχο σχολείο δεν μπορεί να διαγραφεί από τον πίνακα School\_Unit. Η ρήτρα on UPDATE CASCADE σημαίνει ότι εάν ενημερωθεί ένα αναγνωριστικό διαχειριστή στον πίνακα Administrator, το αντίστοιχο αναγνωριστικό σχολείου θα ενημερωθεί επίσης στον πίνακα School\_Unit. Με αυτόν τον τρόπο, ο κώδικας διατηρεί τη συνοχή και την ακεραιότητα των δεδομένων στους πίνακες.

### **Query για την δημιουργία του πίνακα SchoolUser**

```

CREATE TABLE IF NOT EXISTS SchoolUser(
SchoolUserID INT not null,
Position VARCHAR(9) not null check (Position in ('Teacher','Student')),
SchoolID INT not null,
PRIMARY KEY (SchoolUserID),
FOREIGN KEY (SchoolUserID) REFERENCES AppUser(UserID) on DELETE RESTRICT on UPDATE CASCADE,
FOREIGN KEY (SchoolID) REFERENCES School_Unit(SchoolID) on DELETE RESTRICT on UPDATE CASCADE
);

```

SchoolUserID: Πρόκειται για το πρωτεύον κλειδί και το ξένο κλειδί του πίνακα που παραπέμπει στη στήλη UserID του πίνακα AppUser. Αυτό σημαίνει ότι κάθε καθηγητής και μαθητής πρέπει να είναι επίσης χρήστης στον πίνακα AppUser. Ο κώδικας καθορίζει επίσης τις ενέργειες που πρέπει να γίνονται όταν τα δεδομένα στα οποία γίνεται αναφορά τροποποιούνται ή διαγράφονται. Η ρήτρα on DELETE RESTRICT σημαίνει ότι

εάν ένας χρήστης διαγραφεί από τον πίνακα AppUser, ο αντίστοιχος καθηγητής ή μαθητής δεν μπορεί να διαγραφεί από τον πίνακα SchoolUser. Η ρήτρα on UPDATE CASCADE σημαίνει ότι εάν ενημερωθεί ένα αναγνωριστικό χρήστη στον πίνακα AppUser, το αντίστοιχο αναγνωριστικό καθηγητή ή μαθητή θα ενημερωθεί και στον πίνακα SchoolUser. Με αυτόν τον τρόπο, ο κώδικας διατηρεί τη συνοχή και την ακεραιότητα των δεδομένων στους πίνακες.

Position: Αυτή είναι μια συμβολοσειρά έως 9 χαρακτήρων που αποθηκεύει τη θέση του καθηγητή ή του μαθητή. Δεν πρέπει να είναι μηδενική, δηλαδή κάθε καθηγητής ή μαθητής πρέπει να έχει μια θέση. Διαθέτει επίσης έναν περιορισμό ελέγχου που εξασφαλίζει ότι η θέση μπορεί να είναι μόνο είτε "Teacher" είτε "Student", πράγμα που σημαίνει ότι δεν επιτρέπονται άλλες τιμές για αυτή τη στήλη.

SchoolID: Πρόκειται για ξένο κλειδί που παραπέμπει στη στήλη SchoolID του πίνακα School\_Unit. Αυτό σημαίνει ότι κάθε καθηγητής και μαθητής πρέπει να ανήκει σε ένα σχολείο που υπάρχει στον πίνακα School\_Unit. Ο κώδικας καθορίζει επίσης τις ενέργειες που πρέπει να γίνονται όταν τα δεδομένα στα οποία γίνεται αναφορά τροποποιούνται ή διαγράφονται. Η ρήτρα on DELETE RESTRICT σημαίνει ότι εάν ένα σχολείο διαγραφεί από τον πίνακα School\_Unit, οι αντίστοιχοι καθηγητές και μαθητές δεν μπορούν να διαγραφούν από τον πίνακα SchoolUser. Η ρήτρα on UPDATE CASCADE σημαίνει ότι εάν ενημερωθεί ένα αναγνωριστικό σχολείου στον πίνακα School\_Unit, τα αντίστοιχα αναγνωριστικά καθηγητών και μαθητών θα ενημερωθούν επίσης στον πίνακα SchoolUser. Με αυτόν τον τρόπο, ο κώδικας διατηρεί τη συνοχή και την ακεραιότητα των δεδομένων στους πίνακες.

### **Query για την δημιουργία του πίνακα Operator**

```
CREATE TABLE IF NOT EXISTS Operator(  
  OperatorID INT not null,  
  PhoneNumber VARCHAR(45) UNIQUE,  
  AdminID INT not null,  
  SchoolID INT not null,  
  PRIMARY KEY (OperatorID),  
  FOREIGN KEY (OperatorID) REFERENCES AppUser(UserID) on DELETE RESTRICT on UPDATE CASCADE ,  
  FOREIGN KEY (AdminID) REFERENCES Administrator(AdminID) on DELETE RESTRICT on UPDATE CASCADE ,  
  FOREIGN KEY (SchoolID) REFERENCES School_Unit(SchoolID) on DELETE RESTRICT on UPDATE CASCADE  
);
```

OperatorID: Πρόκειται για το πρωτεύον κλειδί και το ξένο κλειδί του πίνακα που παραπέμπει στη στήλη UserID του πίνακα AppUser. Αυτό σημαίνει ότι κάθε χειριστής πρέπει να είναι και χρήστης στον πίνακα AppUser. Ο κώδικας καθορίζει επίσης τις ενέργειες που πρέπει να γίνονται όταν τα δεδομένα στα οποία γίνεται αναφορά τροποποιούνται ή διαγράφονται. Η ρήτρα on DELETE RESTRICT σημαίνει ότι εάν ένας

χρήστης διαγραφεί από τον πίνακα AppUser, ο αντίστοιχος χειριστής δεν μπορεί να διαγραφεί από τον πίνακα Operator. Η ρήτρα on UPDATE CASCADE σημαίνει ότι εάν ενημερωθεί ένα αναγνωριστικό χρήστη στον πίνακα AppUser, το αντίστοιχο αναγνωριστικό χειριστή θα ενημερωθεί και στον πίνακα Operator. Με αυτόν τον τρόπο, ο κώδικας διατηρεί τη συνοχή και την ακεραιότητα των δεδομένων στους πίνακες.

AdminID: Πρόκειται για ένα ξένο κλειδί που παραπέμπει στη στήλη AdminID του πίνακα Administrator. Αυτό σημαίνει ότι κάθε χειριστής πρέπει να εργάζεται για έναν διαχειριστή που υπάρχει στον πίνακα Administrator. Ο κώδικας καθορίζει επίσης τις ενέργειες που πρέπει να γίνονται όταν τα δεδομένα στα οποία γίνεται αναφορά τροποποιούνται ή διαγράφονται. Η ρήτρα on DELETE RESTRICT σημαίνει ότι εάν ένας διαχειριστής διαγραφεί από τον πίνακα Administrator, ο αντίστοιχος χειριστής δεν μπορεί να διαγραφεί από τον πίνακα Operator. Η ρήτρα on UPDATE CASCADE σημαίνει ότι εάν ενημερωθεί ένα αναγνωριστικό διαχειριστή στον πίνακα Administrator, το αντίστοιχο αναγνωριστικό χειριστή θα ενημερωθεί επίσης στον πίνακα Operator. Με αυτόν τον τρόπο, ο κώδικας διατηρεί τη συνοχή και την ακεραιότητα των δεδομένων στους πίνακες.

SchoolID: Πρόκειται για ένα ξένο κλειδί που παραπέμπει στη στήλη SchoolID του πίνακα School\_Unit. Αυτό σημαίνει ότι κάθε χειριστής πρέπει να εργάζεται για ένα σχολείο που υπάρχει στον πίνακα School\_Unit. Ο κωδικός καθορίζει επίσης τις ενέργειες που πρέπει να γίνονται όταν τα δεδομένα στα οποία γίνεται αναφορά τροποποιούνται ή διαγράφονται. Η ρήτρα on DELETE RESTRICT σημαίνει ότι εάν ένα σχολείο διαγραφεί από τον πίνακα School\_Unit, ο αντίστοιχος χειριστής δεν μπορεί να διαγραφεί από τον πίνακα Operator. Η ρήτρα on UPDATE CASCADE σημαίνει ότι εάν ενημερωθεί ένα αναγνωριστικό σχολείου στον πίνακα School\_Unit, το αντίστοιχο αναγνωριστικό χειριστή θα ενημερωθεί επίσης στον πίνακα Operator. Με αυτόν τον τρόπο, ο κώδικας διατηρεί τη συνοχή και την ακεραιότητα των δεδομένων στους πίνακες.

### **Query για την δημιουργία του πίνακα Book**



```

CREATE TABLE IF NOT EXISTS Book(
BookID INT not null AUTO_INCREMENT,
ISBN VARCHAR(17) not null,
Title VARCHAR(60) not null,
Publisher VARCHAR(45),
Number_of_pages INT not null,
Summary TEXT not null,
Image varchar(300) not null,           -- Link for the image
language VARCHAR(45) not null,
OperatorID INT not null,
SchoolID INT not null,
PRIMARY KEY (BookID),
FOREIGN KEY (OperatorID) REFERENCES Operator(OperatorID) on DELETE RESTRICT on UPDATE CASCADE,
FOREIGN KEY (SchoolID) REFERENCES School_Unit(SchoolID) on DELETE RESTRICT on UPDATE CASCADE,
CONSTRAINT UNIQUE (ISBN,SchoolID)
);

```

BookID: Πρόκειται για το πρωτεύον κλειδί του πίνακα που ταυτοποιεί μοναδικά κάθε βιβλίο. Είναι ένας ακέραιος αριθμός που αυξάνεται αυτόματα από το σύστημα της βάσης δεδομένων.

Image: Αυτή είναι μια συμβολοσειρά έως 300 χαρακτήρων που αποθηκεύει έναν σύνδεσμο για την εικόνα του εξωφύλλου του βιβλίου. Δεν πρέπει να είναι μηδενικό, πράγμα που σημαίνει ότι κάθε βιβλίο πρέπει να έχει σύνδεσμο εικόνας. Περιέχει ένα link με την κατάλληλη εικόνα.

OperatorID: Πρόκειται για ξένο κλειδί που παραπέμπει στη στήλη OperatorID του πίνακα Operator. Αυτό σημαίνει ότι κάθε βιβλίο πρέπει να προστίθεται από έναν χειριστή που υπάρχει στον πίνακα Operator. Ο κώδικας καθορίζει επίσης τις ενέργειες που πρέπει να γίνονται όταν τα δεδομένα στα οποία γίνεται αναφορά τροποποιούνται ή διαγράφονται. Η ρήτρα on DELETE RESTRICT σημαίνει ότι εάν ένας χειριστής διαγραφεί από τον πίνακα Operator, το αντίστοιχο βιβλίο δεν μπορεί να διαγραφεί από τον πίνακα Book. Η ρήτρα on UPDATE CASCADE σημαίνει ότι εάν ενημερωθεί ένα αναγνωριστικό χειριστή στον πίνακα Operator, το αντίστοιχο αναγνωριστικό βιβλίου θα ενημερωθεί επίσης στον πίνακα Book. Με αυτόν τον τρόπο, ο κώδικας διατηρεί τη συνοχή και την ακεραιότητα των δεδομένων στους πίνακες.

SchoolID: Πρόκειται για ένα ξένο κλειδί που παραπέμπει στη στήλη SchoolID του πίνακα School\_Unit. Αυτό σημαίνει ότι κάθε βιβλίο πρέπει να ανήκει σε ένα σχολείο που υπάρχει στον πίνακα School\_Unit. Ο κώδικας καθορίζει επίσης τις ενέργειες που πρέπει να γίνονται όταν τα δεδομένα στα οποία γίνεται αναφορά τροποποιούνται ή διαγράφονται. Η ρήτρα on DELETE RESTRICT σημαίνει ότι εάν ένα σχολείο διαγραφεί από τον πίνακα School\_Unit, το αντίστοιχο βιβλίο δεν μπορεί να διαγραφεί από τον πίνακα Book. Η ρήτρα on UPDATE CASCADE σημαίνει ότι εάν ενημερωθεί ένα αναγνωριστικό σχολείου στον πίνακα School\_Unit, το αντίστοιχο αναγνωριστικό βιβλίου θα ενημερωθεί

και στον πίνακα Book. Με αυτόν τον τρόπο, ο κώδικας διατηρεί τη συνοχή και την ακεραιότητα των δεδομένων στους πίνακες.

Ο κώδικας ορίζει επίσης έναν μοναδικό περιορισμό στο συνδυασμό των στηλών ISBN και SchoolID. Αυτό σημαίνει ότι δεν μπορούν δύο βιβλία να έχουν το ίδιο ISBN και να ανήκουν στο ίδιο σχολείο. Αυτό διασφαλίζει ότι δεν υπάρχουν διπλά βιβλία σε ένα σχολείο.

### **Query για την δημιουργία του πίνακα BookCopy**

```
CREATE TABLE IF NOT EXISTS BookCopy(  
  BookCopyID INT not null AUTO_INCREMENT,  
  BookID INT not null,  
  InLibrary BOOLEAN not null DEFAULT TRUE,  
  IsAvailable BOOLEAN not null DEFAULT TRUE,  
  PRIMARY KEY (BookCopyID,BookID),  
  FOREIGN KEY (BookID) REFERENCES Book(BookID) on DELETE RESTRICT on UPDATE CASCADE,  
  CONSTRAINT not_in_library_and_available CHECK ( NOT (InLibrary = FALSE AND IsAvailable = TRUE))  
);
```

BookCopyID: Πρόκειται για το πρωτεύον κλειδί του πίνακα που ταυτοποιεί μοναδικά κάθε αντίγραφο ενός βιβλίου. Είναι ένας ακέραιος αριθμός που αυξάνεται αυτόματα από το σύστημα της βάσης δεδομένων.

BookID: Πρόκειται για το πρωτεύον κλειδί και το ξένο κλειδί του πίνακα που παραπέμπει στη στήλη BookID του πίνακα Book. Αυτό σημαίνει ότι κάθε αντίγραφο ενός βιβλίου πρέπει να ανήκει σε ένα βιβλίο που υπάρχει στον πίνακα Book. Ο κώδικας καθορίζει επίσης τις ενέργειες που πρέπει να γίνονται όταν τα δεδομένα στα οποία γίνεται αναφορά τροποποιούνται ή διαγράφονται. Η ρήτρα on DELETE RESTRICT σημαίνει ότι εάν ένα βιβλίο διαγραφεί από τον πίνακα Book, τα αντίστοιχα αντίγραφα του βιβλίου δεν μπορούν να διαγραφούν από τον πίνακα BookCopy. Η ρήτρα on UPDATE CASCADE σημαίνει ότι εάν ενημερωθεί ένα αναγνωριστικό βιβλίου στον πίνακα Book, τα αντίστοιχα αντίγραφα των αναγνωριστικών βιβλίων θα ενημερωθούν επίσης στον πίνακα BookCopy. Με αυτόν τον τρόπο, ο κώδικας διατηρεί τη συνοχή και την ακεραιότητα των δεδομένων στους πίνακες.

InLibrary: Πρόκειται για μια boolean στήλη που αποθηκεύει αν το αντίγραφο του βιβλίου βρίσκεται στη βιβλιοθήκη ή όχι. Δεν πρέπει να είναι μηδενική, πράγμα που σημαίνει ότι κάθε αντίγραφο ενός βιβλίου πρέπει να έχει μια τιμή για αυτή τη στήλη. Έχει προεπιλεγμένη τιμή TRUE, που σημαίνει ότι αν δεν καθοριστεί τιμή για τη στήλη αυτή, θα θεωρηθεί ότι είναι TRUE.

IsAvailable: Πρόκειται για μια boolean στήλη που αποθηκεύει αν το αντίγραφο του βιβλίου είναι διαθέσιμο για δανεισμό ή όχι. Δεν πρέπει να είναι null, πράγμα που

σημαίνει ότι κάθε αντίγραφο ενός βιβλίου πρέπει να έχει μια τιμή για αυτή τη στήλη. Έχει προεπιλεγμένη τιμή TRUE, που σημαίνει ότι αν δεν καθοριστεί τιμή για τη στήλη αυτή, θα θεωρηθεί ότι είναι TRUE.

Ο κώδικας ορίζει επίσης έναν περιορισμό ελέγχου στο συνδυασμό των στηλών InLibrary και IsAvailable. Αυτό σημαίνει ότι δεν είναι δυνατόν ένα αντίγραφο ενός βιβλίου να μην βρίσκεται στη βιβλιοθήκη και να είναι διαθέσιμο ταυτόχρονα. Αυτό διασφαλίζει ότι δεν υπάρχουν λογικά σφάλματα στα δεδομένα.

### **Query για την δημιουργία του πίνακα Category**

```
CREATE TABLE IF NOT EXISTS Category(  
CategoryID INT not null AUTO_INCREMENT,  
CategoryName VARCHAR(45) not null,  
PRIMARY KEY (CategoryID)  
);
```

CategoryID: Πρόκειται για το πρωτεύον κλειδί του πίνακα που προσδιορίζει μοναδικά κάθε κατηγορία. Είναι ένας ακέραιος αριθμός που αυξάνεται αυτόματα από το σύστημα της βάσης δεδομένων.

CategoryName: Πρόκειται για μια συμβολοσειρά έως 45 χαρακτήρων που αποθηκεύει το όνομα της κατηγορίας. Δεν πρέπει να είναι μηδενικό, πράγμα που σημαίνει ότι κάθε κατηγορία πρέπει να έχει ένα όνομα.

### **Query για την δημιουργία του πίνακα Author**

```
CREATE TABLE IF NOT EXISTS Author(  
AuthorID INT not null AUTO_INCREMENT,  
FullName VARCHAR(90) not null,  
PRIMARY KEY (AuthorID)  
);
```

AuthorID: Πρόκειται για το πρωτεύον κλειδί του πίνακα που ταυτοποιεί μοναδικά κάθε συγγραφέα. Είναι ένας ακέραιος αριθμός που αυξάνεται αυτόματα από το σύστημα της βάσης δεδομένων.

FullName: Πρόκειται για μια συμβολοσειρά έως 90 χαρακτήρων που αποθηκεύει το πλήρες όνομα του συγγραφέα. Δεν πρέπει να είναι μηδενικό, πράγμα που σημαίνει ότι κάθε συγγραφέας πρέπει να έχει όνομα.

### Query για την δημιουργία του πίνακα Keywords

```
CREATE TABLE IF NOT EXISTS Keywords(  
  KeywordID INT not null auto_increment,  
  KeywordText VARCHAR(60) not null,  
  PRIMARY KEY (KeywordID)  
);
```

KeywordID: Πρόκειται για το πρωτεύον κλειδί του πίνακα που προσδιορίζει μοναδικά κάθε λέξη-κλειδί. Είναι ένας ακέραιος αριθμός που αυξάνεται αυτόματα από το σύστημα της βάσης δεδομένων.

KeywordText: Πρόκειται για μια συμβολοσειρά έως 60 χαρακτήρων που αποθηκεύει το κείμενο της λέξης-κλειδί. Δεν πρέπει να είναι μηδενικό, πράγμα που σημαίνει ότι κάθε λέξη-κλειδί πρέπει να έχει κείμενο.

### Query για την δημιουργία του πίνακα BookCategory

```
CREATE TABLE IF NOT EXISTS BookCategory(  
  BookID INT not null,  
  CategoryID INT not null,  
  PRIMARY KEY (BookID,CategoryID),  
  FOREIGN KEY (BookID) REFERENCES Book(BookID) on DELETE RESTRICT on UPDATE CASCADE,  
  FOREIGN KEY (CategoryID) REFERENCES Category(CategoryID) on DELETE RESTRICT on UPDATE CASCADE  
);
```

BookID: Πρόκειται για το πρωτεύον κλειδί και το ξένο κλειδί του πίνακα που παραπέμπει στη στήλη BookID του πίνακα Book. Αυτό σημαίνει ότι κάθε κατηγορία βιβλίου πρέπει να ανήκει σε ένα βιβλίο που υπάρχει στον πίνακα Book. Ο κώδικας καθορίζει επίσης τις ενέργειες που πρέπει να γίνονται όταν τα δεδομένα στα οποία γίνεται αναφορά τροποποιούνται ή διαγράφονται. Η ρήτρα on DELETE RESTRICT σημαίνει ότι αν διαγραφεί ένα βιβλίο από τον πίνακα Book, οι αντίστοιχες κατηγορίες βιβλίων δεν μπορούν να διαγραφούν από τον πίνακα BookCategory. Η ρήτρα on UPDATE CASCADE σημαίνει ότι εάν ενημερωθεί ένα αναγνωριστικό βιβλίου στον πίνακα Book, τα αντίστοιχα αναγνωριστικά κατηγοριών βιβλίων θα ενημερωθούν επίσης στον πίνακα BookCategory. Με αυτόν τον τρόπο, ο κώδικας διατηρεί τη συνοχή και την ακεραιότητα των δεδομένων στους πίνακες.

CategoryID: Πρόκειται για το πρωτεύον κλειδί και το ξένο κλειδί του πίνακα που παραπέμπει στη στήλη CategoryID του πίνακα Category. Αυτό σημαίνει ότι κάθε κατηγορία βιβλίου πρέπει να ανήκει σε μια κατηγορία που υπάρχει στον πίνακα Category. Ο κωδικός καθορίζει επίσης τις ενέργειες που πρέπει να γίνονται όταν τα δεδομένα στα οποία γίνεται αναφορά τροποποιούνται ή διαγράφονται. Η ρήτρα on

DELETE RESTRICT σημαίνει ότι αν διαγραφεί μια κατηγορία από τον πίνακα Category, οι αντίστοιχες κατηγορίες βιβλίων δεν μπορούν να διαγραφούν από τον πίνακα BookCategory. Η ρήτρα on UPDATE CASCADE σημαίνει ότι εάν ενημερωθεί ένα αναγνωριστικό κατηγορίας στον πίνακα Category, τα αντίστοιχα αναγνωριστικά κατηγοριών βιβλίων θα ενημερωθούν επίσης στον πίνακα BookCategory. Με αυτόν τον τρόπο, ο κώδικας διατηρεί τη συνοχή και την ακεραιότητα των δεδομένων στους πίνακες.

Ο κώδικας ορίζει επίσης ένα σύνθετο πρωτεύον κλειδί στο συνδυασμό των στηλών BookID και CategoryID. Αυτό σημαίνει ότι δεν μπορεί να υπάρχουν δύο γραμμές που να έχουν τις ίδιες τιμές και για τις δύο στήλες. Αυτό διασφαλίζει ότι δεν υπάρχουν διπλές κατηγορίες για ένα βιβλίο.

### **Query για την δημιουργία του πίνακα BookAuthor**

```
CREATE TABLE IF NOT EXISTS BookAuthor(  
  BookID INT not null,  
  AuthorID INT not null,  
  PRIMARY KEY (BookID,AuthorID),  
  FOREIGN KEY (BookID) REFERENCES Book(BookID) on DELETE RESTRICT on UPDATE CASCADE,  
  FOREIGN KEY (AuthorID) REFERENCES Author(AuthorID) on DELETE RESTRICT on UPDATE CASCADE  
);
```

BookID: Πρόκειται για το πρωτεύον κλειδί και το ξένο κλειδί του πίνακα που παραπέμπει στη στήλη BookID του πίνακα Book. Αυτό σημαίνει ότι κάθε συγγραφέας βιβλίου πρέπει να ανήκει σε ένα βιβλίο που υπάρχει στον πίνακα Book. Ο κώδικας καθορίζει επίσης τις ενέργειες που πρέπει να γίνονται όταν τα δεδομένα στα οποία γίνεται αναφορά τροποποιούνται ή διαγράφονται. Η ρήτρα on DELETE RESTRICT σημαίνει ότι αν διαγραφεί ένα βιβλίο από τον πίνακα Book, οι αντίστοιχοι συγγραφείς βιβλίων δεν μπορούν να διαγραφούν από τον πίνακα BookAuthor. Η ρήτρα on UPDATE CASCADE σημαίνει ότι εάν ενημερωθεί ένα αναγνωριστικό βιβλίου στον πίνακα Book, τα αντίστοιχα αναγνωριστικά συγγραφέων βιβλίων θα ενημερωθούν επίσης στον πίνακα BookAuthor. Με αυτόν τον τρόπο, ο κώδικας διατηρεί τη συνοχή και την ακεραιότητα των δεδομένων στους πίνακες.

AuthorID: Πρόκειται για το πρωτεύον κλειδί και το ξένο κλειδί του πίνακα που παραπέμπει στη στήλη AuthorID του πίνακα Author. Αυτό σημαίνει ότι κάθε συγγραφέας βιβλίου πρέπει να ανήκει σε έναν συγγραφέα που υπάρχει στον πίνακα Author. Ο κώδικας καθορίζει επίσης τις ενέργειες που πρέπει να γίνονται όταν τα δεδομένα στα οποία γίνεται αναφορά τροποποιούνται ή διαγράφονται. Η ρήτρα on DELETE RESTRICT σημαίνει ότι εάν ένας συγγραφέας διαγραφεί από τον πίνακα Author, οι αντίστοιχοι συγγραφείς βιβλίων δεν μπορούν να διαγραφούν από τον πίνακα BookAuthor. Η ρήτρα

on UPDATE CASCADE σημαίνει ότι εάν ενημερωθεί ένα αναγνωριστικό συγγραφέα στον πίνακα Author, τα αντίστοιχα αναγνωριστικά συγγραφέων βιβλίων θα ενημερωθούν επίσης στον πίνακα BookAuthor. Με αυτόν τον τρόπο, ο κώδικας διατηρεί τη συνοχή και την ακεραιότητα των δεδομένων στους πίνακες.

Ο κώδικας ορίζει επίσης ένα σύνθετο πρωτεύον κλειδί στο συνδυασμό των στηλών BookID και AuthorID. Αυτό σημαίνει ότι δεν μπορεί να υπάρχουν δύο γραμμές που να έχουν τις ίδιες τιμές και για τις δύο στήλες. Αυτό διασφαλίζει ότι δεν υπάρχουν διπλοί συγγραφείς για ένα βιβλίο.

### **Query για την δημιουργία του πίνακα BookKeywords**

```
CREATE TABLE IF NOT EXISTS BookKeywords(  
BookID INT not null,  
KeywordID INT not null,  
PRIMARY KEY (BookID,KeywordID),  
FOREIGN KEY (BookID) REFERENCES Book(BookID) on DELETE RESTRICT on UPDATE CASCADE,  
FOREIGN KEY (KeywordID) REFERENCES Keywords(KeywordID) on DELETE RESTRICT on UPDATE CASCADE  
);
```

BookID: Πρόκειται για το πρωτεύον κλειδί και το ξένο κλειδί του πίνακα που παραπέμπει στη στήλη BookID του πίνακα Book. Αυτό σημαίνει ότι κάθε λέξη-κλειδί βιβλίο πρέπει να ανήκει σε ένα βιβλίο που υπάρχει στον πίνακα Book. Ο κώδικας καθορίζει επίσης τις ενέργειες που πρέπει να γίνονται όταν τα δεδομένα στα οποία γίνεται αναφορά τροποποιούνται ή διαγράφονται. Η ρήτρα on DELETE RESTRICT σημαίνει ότι εάν ένα βιβλίο διαγραφεί από τον πίνακα Book, οι αντίστοιχες λέξεις-κλειδιά βιβλίου δεν μπορούν να διαγραφούν από τον πίνακα BookKeywords. Η ρήτρα on UPDATE CASCADE σημαίνει ότι εάν ενημερωθεί ένα αναγνωριστικό βιβλίου στον πίνακα Book, τα αντίστοιχα αναγνωριστικά λέξεων-κλειδιών βιβλίου θα ενημερωθούν επίσης στον πίνακα BookKeywords. Με αυτόν τον τρόπο, ο κώδικας διατηρεί τη συνοχή και την ακεραιότητα των δεδομένων στους πίνακες.

KeywordID: Πρόκειται για το πρωτεύον κλειδί και το ξένο κλειδί του πίνακα που παραπέμπει στη στήλη KeywordID του πίνακα Keywords. Αυτό σημαίνει ότι κάθε λέξη-κλειδί βιβλίου πρέπει να ανήκει σε μια λέξη-κλειδί που υπάρχει στον πίνακα Keywords. Ο κώδικας καθορίζει επίσης τις ενέργειες που πρέπει να γίνονται όταν τα δεδομένα στα οποία γίνεται αναφορά τροποποιούνται ή διαγράφονται. Η ρήτρα on DELETE RESTRICT σημαίνει ότι αν διαγραφεί μια λέξη-κλειδί από τον πίνακα Keywords, οι αντίστοιχες λέξεις-κλειδιά βιβλίων δεν μπορούν να διαγραφούν από τον πίνακα BookKeywords. Η ρήτρα on UPDATE CASCADE σημαίνει ότι εάν ενημερωθεί ένα αναγνωριστικό λέξης-κλειδιού στον πίνακα Keywords, τα αντίστοιχα αναγνωριστικά λέξεων-κλειδιών βιβλίου



θα ενημερωθούν επίσης στον πίνακα BookKeywords. Με αυτόν τον τρόπο, ο κώδικας διατηρεί τη συνοχή και την ακεραιότητα των δεδομένων στους πίνακες.

Ο κώδικας ορίζει επίσης ένα σύνθετο πρωτεύον κλειδί στο συνδυασμό των στηλών BookID και KeywordID. Αυτό σημαίνει ότι δεν μπορεί να υπάρχουν δύο γραμμές που να έχουν τις ίδιες τιμές και για τις δύο στήλες. Αυτό διασφαλίζει ότι δεν υπάρχουν διπλές λέξεις-κλειδιά για ένα βιβλίο.

### **Query για την δημιουργία του πίνακα Borrow**

```
CREATE TABLE IF NOT EXISTS Borrow(  
  BorrowID INT not null AUTO_INCREMENT,  
  SchoolUserID INT not null,  
  BookID INT not null,  
  BookCopyID INT not null,  
  Borrow_Date DATE not null DEFAULT (CURRENT_DATE),  
  Due_Date DATE not null DEFAULT (DATE_ADD(CURRENT_DATE,INTERVAL 1 WEEK)),  
  Return_Date DATE,  
  Borrow_Status VARCHAR(12) not null DEFAULT 'Accepted' CHECK (Borrow_Status in ('Accepted','Overdue','Lost','Returned')),  
  OperatorID INT not null,  
  PRIMARY KEY (BorrowID, SchoolUserID, BookID),  
  FOREIGN KEY (SchoolUserID) REFERENCES SchoolUser(SchoolUserID) on DELETE RESTRICT on UPDATE CASCADE,  
  FOREIGN KEY (BookID) REFERENCES Book(BookID) on DELETE RESTRICT on UPDATE CASCADE,  
  FOREIGN KEY (BookCopyID) REFERENCES BookCopy(BookCopyID) on DELETE RESTRICT on UPDATE CASCADE ,  
  FOREIGN KEY (OperatorID) REFERENCES Operator(OperatorID) on DELETE RESTRICT on UPDATE CASCADE  
);
```

BorrowID: Πρόκειται για το πρωτεύον κλειδί του πίνακα που ταυτοποιεί μοναδικά κάθε δανεισμό. Είναι ένας ακέραιος αριθμός που αυξάνεται αυτόματα από το σύστημα της βάσης δεδομένων.

SchoolUserID: Πρόκειται για το πρωτεύον κλειδί και το ξένο κλειδί του πίνακα που παραπέμπει στη στήλη SchoolUserID του πίνακα SchoolUser. Αυτό σημαίνει ότι κάθε δανεισμός πρέπει να ανήκει σε έναν καθηγητή ή έναν μαθητή που υπάρχει στον πίνακα SchoolUser. Ο κωδικός καθορίζει επίσης τις ενέργειες που πρέπει να γίνονται όταν τα δεδομένα στα οποία γίνεται αναφορά τροποποιούνται ή διαγράφονται. Η ρήτρα on DELETE RESTRICT σημαίνει ότι εάν ένας καθηγητής ή ένας μαθητής διαγραφεί από τον πίνακα SchoolUser, τα αντίστοιχα δάνεια δεν μπορούν να διαγραφούν από τον πίνακα Borrow. Η ρήτρα on UPDATE CASCADE σημαίνει ότι εάν ενημερωθεί ένα αναγνωριστικό καθηγητή ή μαθητή στον πίνακα SchoolUser, τα αντίστοιχα αναγνωριστικά δανείων θα ενημερωθούν και στον πίνακα Borrow. Με αυτόν τον τρόπο, ο κώδικας διατηρεί τη συνοχή και την ακεραιότητα των δεδομένων στους πίνακες.

BookID: Πρόκειται για το πρωτεύον κλειδί και το ξένο κλειδί του πίνακα που παραπέμπει στη στήλη BookID του πίνακα Book. Αυτό σημαίνει ότι κάθε δανεισμός πρέπει να ανήκει σε ένα βιβλίο που υπάρχει στον πίνακα Book. Ο κώδικας καθορίζει επίσης τις ενέργειες που πρέπει να γίνονται όταν τα δεδομένα στα οποία γίνεται αναφορά τροποποιούνται ή

διαγράφονται. Η ρήτρα on DELETE RESTRICT σημαίνει ότι αν διαγραφεί ένα βιβλίο από τον πίνακα Book, τα αντίστοιχα δανειστικά στοιχεία δεν μπορούν να διαγραφούν από τον πίνακα Borrow. Η ρήτρα on UPDATE CASCADE σημαίνει ότι εάν ενημερωθεί ένα αναγνωριστικό βιβλίου στον πίνακα Book, τα αντίστοιχα αναγνωριστικά δανείων θα ενημερωθούν επίσης στον πίνακα Borrow. Με αυτόν τον τρόπο, ο κώδικας διατηρεί τη συνοχή και την ακεραιότητα των δεδομένων στους πίνακες.

BookCopyID: Πρόκειται για ένα ξένο κλειδί που παραπέμπει στη στήλη BookCopyID του πίνακα BookCopy. Αυτό σημαίνει ότι κάθε δανεισμός πρέπει να ανήκει σε ένα αντίγραφο ενός βιβλίου που υπάρχει στον πίνακα BookCopy. Ο κώδικας καθορίζει επίσης τις ενέργειες που πρέπει να γίνονται όταν τα δεδομένα στα οποία γίνεται αναφορά τροποποιούνται ή διαγράφονται. Η ρήτρα on DELETE RESTRICT σημαίνει ότι εάν διαγραφεί ένα αντίγραφο ενός βιβλίου από τον πίνακα BookCopy, ο αντίστοιχος δανεισμός δεν μπορεί να διαγραφεί από τον πίνακα Borrow. Η ρήτρα on UPDATE CASCADE σημαίνει ότι εάν ενημερωθεί ένα αντίγραφο ενός αναγνωριστικού βιβλίου στον πίνακα BookCopy, το αντίστοιχο αναγνωριστικό δανεισμού θα ενημερωθεί επίσης στον πίνακα Borrow. Με αυτόν τον τρόπο, ο κώδικας διατηρεί τη συνοχή και την ακεραιότητα των δεδομένων στους πίνακες.

Borrow\_Date: Δανεισμός: Αυτή είναι μια στήλη ημερομηνίας που αποθηκεύει την ημερομηνία κατά την οποία έγινε ο δανεισμός. Δεν πρέπει να είναι μηδενική, πράγμα που σημαίνει ότι κάθε δανεισμός πρέπει να έχει ημερομηνία δανεισμού. Έχει προεπιλεγμένη τιμή CURRENT\_DATE, που σημαίνει ότι αν δεν καθοριστεί τιμή για τη στήλη αυτή, θα θεωρηθεί ότι είναι η τρέχουσα ημερομηνία.

Due\_Date: Πρόκειται για μια στήλη ημερομηνίας που αποθηκεύει την ημερομηνία κατά την οποία πρέπει να επιστραφεί ο δανεισμός. Δεν πρέπει να είναι μηδενική, πράγμα που σημαίνει ότι κάθε δανεισμός πρέπει να έχει ημερομηνία λήξης. Έχει προεπιλεγμένη τιμή DATE\_ADD(CURRENT\_DATE,INTERVAL 1 WEEK), που σημαίνει ότι εάν δεν καθοριστεί τιμή για τη στήλη αυτή, θα θεωρηθεί ότι είναι μία εβδομάδα μετά την τρέχουσα ημερομηνία.

Return\_Date: Πρόκειται για μια στήλη ημερομηνίας που αποθηκεύει την ημερομηνία κατά την οποία επιστράφηκε ο δανεισμός. Μπορεί να είναι μηδενική, πράγμα που σημαίνει ότι ορισμένα δάνεια ενδέχεται να μην έχουν ακόμη ημερομηνία επιστροφής.

Borrow\_Status: Πρόκειται για μια συμβολοσειρά έως 12 χαρακτήρων που αποθηκεύει την κατάσταση του δανεισμού. Δεν πρέπει να είναι null, που σημαίνει ότι κάθε δανεισμός πρέπει να έχει μια κατάσταση. Έχει προεπιλεγμένη τιμή "Accepted", που σημαίνει ότι αν δεν καθοριστεί τιμή για τη στήλη αυτή, θα θεωρηθεί ότι είναι "Accepted". Διαθέτει επίσης έναν περιορισμό ελέγχου που εξασφαλίζει ότι η κατάσταση μπορεί να είναι μόνο μία από τις τιμές "Αποδεκτός", "Καθυστερημένος", "Απολεσθείς" ή



"Επιστραφείς", πράγμα που σημαίνει ότι δεν επιτρέπονται άλλες τιμές για τη στήλη αυτή.

OperatorID: Πρόκειται για ξένο κλειδί που παραπέμπει στη στήλη OperatorID του πίνακα Operator. Αυτό σημαίνει ότι κάθε δανεισμός πρέπει να γίνεται από έναν χειριστή που υπάρχει στον πίνακα Operator. Ο κώδικας καθορίζει επίσης τις ενέργειες που πρέπει να γίνονται όταν τα δεδομένα στα οποία γίνεται αναφορά τροποποιούνται ή διαγράφονται. Η ρήτρα on DELETE RESTRICT σημαίνει ότι εάν ένας χειριστής διαγραφεί από τον πίνακα Operator, ο αντίστοιχος δανεισμός δεν μπορεί να διαγραφεί από τον πίνακα Borrow. Η ρήτρα on UPDATE CASCADE σημαίνει ότι εάν ενημερωθεί ένα αναγνωριστικό φορέα στον πίνακα Operator, το αντίστοιχο αναγνωριστικό δανείου θα ενημερωθεί και στον πίνακα Borrow. Με αυτόν τον τρόπο, ο κώδικας διατηρεί τη συνοχή και την ακεραιότητα των δεδομένων στους πίνακες.

### **Query για την δημιουργία του πίνακα Reservation**

```
CREATE TABLE IF NOT EXISTS Reservation(  
ReservationID INT not null AUTO_INCREMENT,  
SchoolUserID INT not null,  
BookID INT not null,  
BookCopyID INT,  
Submitted_DateTime DATETIME not null DEFAULT (CURRENT_TIMESTAMP),  
Acceptance_Date DATE,  
Due_Reservation_Date DATE,  
Cancellation_Date DATE CHECK (Cancellation_Date >= Acceptance_Date),  
ReservationStatus VARCHAR(12) not null DEFAULT 'Pending' CHECK (ReservationStatus in ('Pending','Cancelled','Accepted','Completed')),  
OperatorID INT not null,  
PRIMARY KEY (ReservationID, SchoolUserID, BookID),  
FOREIGN KEY (SchoolUserID) REFERENCES SchoolUser(SchoolUserID) on DELETE RESTRICT on UPDATE CASCADE,  
FOREIGN KEY (BookID) REFERENCES Book(BookID) on DELETE RESTRICT on UPDATE CASCADE,  
FOREIGN KEY (BookCopyID) REFERENCES BookCopy(BookCopyID) on DELETE RESTRICT on UPDATE CASCADE,  
FOREIGN KEY (OperatorID) REFERENCES Operator(OperatorID) on DELETE RESTRICT on UPDATE CASCADE  
);
```

ReservationID: Πρόκειται για το πρωτεύον κλειδί του πίνακα που ταυτοποιεί μοναδικά κάθε κράτηση. Είναι ένας ακέραιος αριθμός που αυξάνεται αυτόματα από το σύστημα της βάσης δεδομένων.

SchoolUserID: Πρόκειται για το πρωτεύον κλειδί και το ξένο κλειδί του πίνακα που παραπέμπει στη στήλη SchoolUserID του πίνακα SchoolUser. Αυτό σημαίνει ότι κάθε κράτηση πρέπει να ανήκει σε καθηγητή ή μαθητή που υπάρχει στον πίνακα SchoolUser. Ο κωδικός καθορίζει επίσης τις ενέργειες που πρέπει να γίνονται όταν τα δεδομένα στα οποία γίνεται αναφορά τροποποιούνται ή διαγράφονται. Η ρήτρα on DELETE RESTRICT σημαίνει ότι εάν ένας καθηγητής ή ένας μαθητής διαγραφεί από τον πίνακα SchoolUser, οι αντίστοιχες κρατήσεις δεν μπορούν να διαγραφούν από τον πίνακα Reservation. Η ρήτρα on UPDATE CASCADE σημαίνει ότι εάν ενημερωθεί ένα αναγνωριστικό καθηγητή ή μαθητή στον πίνακα SchoolUser, τα αντίστοιχα αναγνωριστικά κρατήσεων θα

ενημερωθούν επίσης στον πίνακα Reservation. Με αυτόν τον τρόπο, ο κώδικας διατηρεί τη συνοχή και την ακεραιότητα των δεδομένων στους πίνακες.

**BookID:** Πρόκειται για το πρωτεύον κλειδί και το ξένο κλειδί του πίνακα που παραπέμπει στη στήλη BookID του πίνακα Book. Αυτό σημαίνει ότι κάθε κράτηση πρέπει να ανήκει σε ένα βιβλίο που υπάρχει στον πίνακα Book. Ο κώδικας καθορίζει επίσης τις ενέργειες που πρέπει να γίνονται όταν τα δεδομένα στα οποία γίνεται αναφορά τροποποιούνται ή διαγράφονται. Η ρήτρα on DELETE RESTRICT σημαίνει ότι αν διαγραφεί ένα βιβλίο από τον πίνακα Book, οι αντίστοιχες κρατήσεις δεν μπορούν να διαγραφούν από τον πίνακα Reservation. Η ρήτρα on UPDATE CASCADE σημαίνει ότι εάν ενημερωθεί ένα αναγνωριστικό βιβλίου στον πίνακα Book, τα αντίστοιχα αναγνωριστικά κρατήσεων θα ενημερωθούν επίσης στον πίνακα Reservation. Με αυτόν τον τρόπο, ο κώδικας διατηρεί τη συνοχή και την ακεραιότητα των δεδομένων στους πίνακες.

**BookCopyID:** Πρόκειται για ένα ξένο κλειδί που παραπέμπει στη στήλη BookCopyID του πίνακα BookCopy. Αυτό σημαίνει ότι κάθε κράτηση μπορεί να ανήκει σε ένα αντίγραφο ενός βιβλίου που υπάρχει στον πίνακα BookCopy. Ο κώδικας καθορίζει επίσης τις ενέργειες που πρέπει να γίνονται όταν τα δεδομένα στα οποία γίνεται αναφορά τροποποιούνται ή διαγράφονται. Η ρήτρα on DELETE RESTRICT σημαίνει ότι εάν διαγραφεί ένα αντίγραφο ενός βιβλίου από τον πίνακα BookCopy, η αντίστοιχη κράτηση δεν μπορεί να διαγραφεί από τον πίνακα Reservation. Η ρήτρα on UPDATE CASCADE σημαίνει ότι εάν ενημερωθεί ένα αντίγραφο ενός αναγνωριστικού βιβλίου στον πίνακα BookCopy, το αντίστοιχο αναγνωριστικό κράτησης θα ενημερωθεί επίσης στον πίνακα Reservation. Με αυτόν τον τρόπο, ο κώδικας διατηρεί τη συνοχή και την ακεραιότητα των δεδομένων στους πίνακες.

**Submitted\_DateTime:** Πρόκειται για μια στήλη datetime που αποθηκεύει την ημερομηνία και την ώρα κατά την οποία υποβλήθηκε η κράτηση. Δεν πρέπει να είναι μηδενική, πράγμα που σημαίνει ότι κάθε κράτηση πρέπει να έχει ημερομηνία και ώρα υποβολής. Έχει προεπιλεγμένη τιμή CURRENT\_TIMESTAMP, που σημαίνει ότι εάν δεν καθοριστεί τιμή για τη στήλη αυτή, θα θεωρηθεί ότι είναι η τρέχουσα ημερομηνία και ώρα.

**Acceptance\_Date:** Αυτή είναι μια στήλη ημερομηνίας που αποθηκεύει την ημερομηνία κατά την οποία η κράτηση έγινε αποδεκτή από έναν χειριστή. Μπορεί να είναι μηδενική, πράγμα που σημαίνει ότι ορισμένες κρατήσεις μπορεί να μην έχουν ακόμη ημερομηνία αποδοχής.

**Due\_Reservation\_Date:** Πρόκειται για μια στήλη ημερομηνίας που αποθηκεύει την ημερομηνία κατά την οποία η κράτηση πρέπει να ολοκληρωθεί με δανεισμό ή ακύρωση. Μπορεί να είναι μηδενική, πράγμα που σημαίνει ότι ορισμένες κρατήσεις ενδέχεται να μην έχουν ακόμη ημερομηνία λήξης κράτησης.

Cancelation\_Date: Πρόκειται για μια στήλη ημερομηνίας που αποθηκεύει την ημερομηνία κατά την οποία η κράτηση ακυρώθηκε από έναν χειριστή ή έναν καθηγητή ή έναν μαθητή. Μπορεί να είναι μηδενική, πράγμα που σημαίνει ότι ορισμένες κρατήσεις ενδέχεται να μην έχουν ακόμη ημερομηνία ακύρωσης. Διαθέτει επίσης έναν περιορισμό ελέγχου που εξασφαλίζει ότι η ημερομηνία ακύρωσης δεν μπορεί να είναι προγενέστερη από την ημερομηνία αποδοχής, πράγμα που σημαίνει ότι καμία κράτηση δεν μπορεί να ακυρωθεί πριν γίνει αποδεκτή.

Κατάσταση κράτησης: Πρόκειται για μια συμβολοσειρά έως 12 χαρακτήρων που αποθηκεύει την κατάσταση της κράτησης. Δεν πρέπει να είναι μηδέν, πράγμα που σημαίνει ότι κάθε κράτηση πρέπει να έχει μια κατάσταση. Έχει προεπιλεγμένη τιμή "Pending", που σημαίνει ότι εάν δεν καθοριστεί τιμή για τη στήλη αυτή, θα θεωρηθεί ότι είναι "Pending". Διαθέτει επίσης έναν περιορισμό ελέγχου που εξασφαλίζει ότι η κατάσταση μπορεί να είναι μόνο μία από τις τιμές "Pending", "Cancelled", "Accepted" ή "Completed", πράγμα που σημαίνει ότι δεν επιτρέπονται άλλες τιμές για αυτή τη στήλη.

OperatorID: Πρόκειται για ξένο κλειδί που παραπέμπει στη στήλη OperatorID του πίνακα Operator. Αυτό σημαίνει ότι κάθε κράτηση πρέπει να γίνεται από έναν χειριστή που υπάρχει στον πίνακα Operator. Ο κώδικας καθορίζει επίσης τις ενέργειες που πρέπει να γίνονται όταν τα δεδομένα στα οποία γίνεται αναφορά τροποποιούνται ή διαγράφονται. Η ρήτρα on DELETE RESTRICT σημαίνει ότι εάν ένας χειριστής διαγραφεί από τον πίνακα Operator, η αντίστοιχη κράτηση δεν μπορεί να διαγραφεί από τον πίνακα Reservation. Η ρήτρα on UPDATE CASCADE σημαίνει ότι εάν ενημερωθεί ένα αναγνωριστικό φορέα εκμετάλλευσης στον πίνακα Operator, το αντίστοιχο αναγνωριστικό κράτησης θα ενημερωθεί επίσης στον πίνακα Reservation. Με αυτόν τον τρόπο, ο κώδικας διατηρεί τη συνοχή και την ακεραιότητα των δεδομένων στους πίνακες.

### **Query για την δημιουργία του πίνακα Review**

```
CREATE TABLE IF NOT EXISTS Review(  
ReviewID INT not null AUTO_INCREMENT,  
SchoolUserID INT not null,  
BookID INT not null,  
Review_text TEXT,  
Likert INT not null DEFAULT 0 CHECK ( Likert BETWEEN 1 AND 5),  
Date_Submitted DATETIME not null DEFAULT CURRENT_TIMESTAMP,  
Approval_Status VARCHAR(45) not null DEFAULT 'Pending' CHECK (Approval_Status in ( 'Pending','Approved','Rejected')) ,  
OperatorID INT not null,  
PRIMARY KEY ( ReviewID, SchoolUserID, BookID ),  
FOREIGN KEY (SchoolUserID) REFERENCES SchoolUser(SchoolUserID) on DELETE RESTRICT on UPDATE CASCADE,  
FOREIGN KEY (BookID) REFERENCES Book(BookID) on DELETE RESTRICT on UPDATE CASCADE,  
FOREIGN KEY (OperatorID) REFERENCES Operator(OperatorID) on DELETE RESTRICT on UPDATE CASCADE,  
CONSTRAINT unique_review_per_book_by_user UNIQUE (BookID, SchoolUserID)  
);
```

**ReviewID:** Πρόκειται για το πρωτεύον κλειδί του πίνακα που προσδιορίζει μοναδικά κάθε κριτική. Είναι ένας ακεραίος αριθμός που αυξάνεται αυτόματα από το σύστημα της βάσης δεδομένων.

**SchoolUserID:** Πρόκειται για το πρωτεύον κλειδί και το ξένο κλειδί του πίνακα που παραπέμπει στη στήλη SchoolUserID του πίνακα SchoolUser. Αυτό σημαίνει ότι κάθε κριτική πρέπει να ανήκει σε έναν καθηγητή ή έναν μαθητή που υπάρχει στον πίνακα SchoolUser. Ο κωδικός καθορίζει επίσης τις ενέργειες που πρέπει να γίνονται όταν τα δεδομένα στα οποία γίνεται αναφορά τροποποιούνται ή διαγράφονται. Η ρήτρα on DELETE RESTRICT σημαίνει ότι εάν ένας καθηγητής ή ένας μαθητής διαγραφεί από τον πίνακα SchoolUser, οι αντίστοιχες κριτικές δεν μπορούν να διαγραφούν από τον πίνακα Review. Η ρήτρα on UPDATE CASCADE σημαίνει ότι αν ενημερωθεί ένα αναγνωριστικό καθηγητή ή μαθητή στον πίνακα SchoolUser, τα αντίστοιχα αναγνωριστικά κριτικών θα ενημερωθούν και στον πίνακα Review. Με αυτόν τον τρόπο, ο κώδικας διατηρεί τη συνοχή και την ακεραιότητα των δεδομένων στους πίνακες.

**BookID:** Πρόκειται για το πρωτεύον κλειδί και το ξένο κλειδί του πίνακα που παραπέμπει στη στήλη BookID του πίνακα Book. Αυτό σημαίνει ότι κάθε κριτική πρέπει να ανήκει σε ένα βιβλίο που υπάρχει στον πίνακα Book. Ο κώδικας καθορίζει επίσης τις ενέργειες που πρέπει να γίνονται όταν τα δεδομένα στα οποία γίνεται αναφορά τροποποιούνται ή διαγράφονται. Η ρήτρα on DELETE RESTRICT σημαίνει ότι εάν ένα βιβλίο διαγραφεί από τον πίνακα Book, οι αντίστοιχες κριτικές δεν μπορούν να διαγραφούν από τον πίνακα Review. Η ρήτρα on UPDATE CASCADE σημαίνει ότι εάν ενημερωθεί ένα αναγνωριστικό βιβλίου στον πίνακα Book, τα αντίστοιχα αναγνωριστικά κριτικών θα ενημερωθούν επίσης στον πίνακα Review. Με αυτόν τον τρόπο, ο κώδικας διατηρεί τη συνοχή και την ακεραιότητα των δεδομένων στους πίνακες.

**Review\_text:** Πρόκειται για μια στήλη κειμένου που αποθηκεύει το κείμενο της κριτικής. Μπορεί να είναι μηδενική, πράγμα που σημαίνει ότι ορισμένες κριτικές μπορεί να μην έχουν κείμενο.

**Likert:** Πρόκειται για μια ακεραία στήλη που αποθηκεύει τη βαθμολογία της κριτικής σε μια κλίμακα από το 1 έως το 5, όπου το 1 είναι πολύ κακό και το 5 είναι πολύ καλό. Δεν πρέπει να είναι μηδενική, πράγμα που σημαίνει ότι κάθε κριτική πρέπει να έχει βαθμολογία. Έχει προεπιλεγμένη τιμή 0, πράγμα που σημαίνει ότι αν δεν καθοριστεί τιμή για αυτή τη στήλη, θα θεωρηθεί ότι είναι 0. Έχει επίσης έναν περιορισμό ελέγχου που εξασφαλίζει ότι η βαθμολογία μπορεί να είναι μόνο μεταξύ 1 και 5, πράγμα που σημαίνει ότι δεν επιτρέπονται άλλες τιμές για αυτή τη στήλη.

**Date\_Submitted:** Πρόκειται για μια στήλη datetime που αποθηκεύει την ημερομηνία και την ώρα κατά την οποία υποβλήθηκε η κριτική. Δεν πρέπει να είναι μηδενική, πράγμα που σημαίνει ότι κάθε κριτική πρέπει να έχει ημερομηνία και ώρα υποβολής. Έχει

προεπιλεγμένη τιμή CURRENT\_TIMESTAMP, που σημαίνει ότι εάν δεν καθοριστεί τιμή για τη στήλη αυτή, θα θεωρηθεί ότι είναι η τρέχουσα ημερομηνία και ώρα.

Approval\_Status: Αυτή είναι μια συμβολοσειρά έως 45 χαρακτήρων που αποθηκεύει την κατάσταση της έγκρισης της αναθεώρησης από έναν χειριστή. Δεν πρέπει να είναι μηδενική, πράγμα που σημαίνει ότι κάθε επισκόπηση πρέπει να έχει κατάσταση έγκρισης. Έχει προεπιλεγμένη τιμή "Pending", που σημαίνει ότι αν δεν καθοριστεί τιμή για τη στήλη αυτή, θα θεωρηθεί ότι είναι "Pending". Διαθέτει επίσης έναν περιορισμό ελέγχου που εξασφαλίζει ότι η κατάσταση έγκρισης μπορεί να είναι μόνο μία από τις τιμές "Εκκρεμής", "Εγκρίθηκε" ή "Απορρίφθηκε", πράγμα που σημαίνει ότι δεν επιτρέπονται άλλες τιμές για αυτή τη στήλη.

OperatorID: Πρόκειται για ξένο κλειδί που παραπέμπει στη στήλη OperatorID του πίνακα Operator. Αυτό σημαίνει ότι κάθε αναθεώρηση πρέπει να εγκρίνεται από έναν χειριστή που υπάρχει στον πίνακα Operator. Ο κωδικός καθορίζει επίσης τις ενέργειες που πρέπει να γίνονται όταν τα δεδομένα στα οποία γίνεται αναφορά τροποποιούνται ή διαγράφονται. Η ρήτρα on DELETE RESTRICT σημαίνει ότι εάν ένας χειριστής διαγραφεί από τον πίνακα Operator, η αντίστοιχη επισκόπηση δεν μπορεί να διαγραφεί από τον πίνακα Review. Η ρήτρα on UPDATE CASCADE σημαίνει ότι εάν ενημερωθεί ένα αναγνωριστικό φορέα στον πίνακα Operator, το αντίστοιχο αναγνωριστικό επισκόπησης θα ενημερωθεί και στον πίνακα Review. Με αυτόν τον τρόπο, ο κώδικας διατηρεί τη συνοχή και την ακεραιότητα των δεδομένων στους πίνακες.

Ο κώδικας ορίζει επίσης έναν μοναδικό περιορισμό στο συνδυασμό των στηλών BookID και SchoolUserID. Αυτό σημαίνει ότι δεν μπορεί να υπάρχουν δύο κριτικές που να έχουν το ίδιο βιβλίο και τον ίδιο καθηγητή ή μαθητή. Αυτό διασφαλίζει ότι δεν υπάρχουν διπλές κριτικές για ένα βιβλίο από έναν καθηγητή ή έναν μαθητή.

### **Query για την δημιουργία του πίνακα RegistrationRequest**

```

CREATE TABLE IF NOT EXISTS RegistrationRequest(
RequestID INT not null AUTO_INCREMENT,
FirstName VARCHAR(45) not null,
LastName VARCHAR(45) not null,
BirthDate DATE not null,
Email VARCHAR(80) UNIQUE not null,
PhoneNumber VARCHAR(45) UNIQUE,
Username VARCHAR(80) UNIQUE not null,
Password VARCHAR(100) not null,
School_Unit INT not null,
Role_Type VARCHAR(20) not null CHECK (Role_Type in ('operator','student','teacher') ),
PRIMARY KEY (RequestID)
);

```

RequestID: Πρόκειται για το πρωτεύον κλειδί του πίνακα που ταυτοποιεί μοναδικά κάθε αίτηση. Είναι ένας ακέραιος αριθμός που αυξάνεται αυτόματα από το σύστημα της βάσης δεδομένων.

School\_Unit: Πρόκειται για μια ακέραια στήλη που αποθηκεύει το αναγνωριστικό της σχολικής μονάδας στην οποία θέλει να ενταχθεί ο χρήστης. Δεν πρέπει να είναι μηδενική, πράγμα που σημαίνει ότι κάθε χρήστης πρέπει να επιλέξει μια σχολική μονάδα.

Role\_Type: Πρόκειται για μια συμβολοσειρά έως 20 χαρακτήρων που αποθηκεύει τον τύπο ρόλου του χρήστη. Δεν πρέπει να είναι μηδέν, πράγμα που σημαίνει ότι κάθε χρήστης πρέπει να επιλέξει έναν τύπο ρόλου. Διαθέτει επίσης έναν περιορισμό ελέγχου που εξασφαλίζει ότι ο τύπος ρόλου μπορεί να είναι μόνο ένας από τους τύπους "operator", "student" ή "teacher", πράγμα που σημαίνει ότι δεν επιτρέπονται άλλες τιμές για αυτή τη στήλη.

### **Query για την δημιουργία των indeces**

```
-- INDEXES
```

```

CREATE INDEX idx_login ON AppUser(Username,Password);
CREATE INDEX idx_schooluserInfo ON SchoolUser(SchoolID,SchoolUserID,Position);
CREATE INDEX idx_book_title_School ON Book(BookID,Title,SchoolID);
CREATE INDEX idx_category ON Category(CategoryName);
CREATE INDEX idx_borrow ON Borrow(SchoolUserID, Borrow_Date, BorrowID, BookID);
CREATE INDEX idx_review ON Review(SchoolUserID, BookID, OperatorID);

```

Αυτός ο κώδικας δημιουργεί έξι ευρετήρια. Ένα ευρετήριο είναι μια δομή δεδομένων που βελτιώνει την ταχύτητα των λειτουργιών ανάκτησης δεδομένων σε έναν πίνακα της

βάσης δεδομένων. Το πρώτο ευρετήριο ονομάζεται `idx_login` και δημιουργείται στον πίνακα `AppUser` με τις στήλες `Username` και `Password`. Ο δεύτερος δείκτης ονομάζεται `idx_schooluserInfo` και δημιουργείται στον πίνακα `SchoolUser` με τις στήλες `SchoolID`, `SchoolUserID` και `Position`. Ο τρίτος δείκτης ονομάζεται `idx_book_title_School` και δημιουργείται στον πίνακα `Book` με τις στήλες `BookID`, `Title` και `SchoolID`. Ο τέταρτος δείκτης ονομάζεται `idx_category` και δημιουργείται στον πίνακα `Category` με τη στήλη `CategoryName`. Ο πέμπτος δείκτης ονομάζεται `idx_borrow` και δημιουργείται στον πίνακα `Borrow` με τις στήλες `SchoolUserID`, `Borrow_Date`, `BorrowID` και `BookID`. Τέλος, ο έκτος δείκτης ονομάζεται `idx_review` και δημιουργείται στον πίνακα `Review` με τις στήλες `SchoolUserID`, `BookID` και `OperatorID`.

## TRIGGERS – EVENTS - PROCEDURES

Απαραίτητος είναι και ο φάκελος `TRIGGERS_EVENTS_PROCEDURES.sql`, που περιέχει τον κώδικα `sql`, που χρειάζεται για δημιουργήσει αντίστοιχα `triggers`, `events` and `procedures`, που κάνουν την βάση πιο ασφαλή και τις λειτουργίες του συστήματος βιβλιοθηκών, που φτιάχνουμε ευκολότερες και ταχύτερες

## DML Script

Ο φάκελος `DML_SCRIPT.sql` περιέχει τον `SQL` κώδικα για την εισαγωγή των `dummy data` στην βάση. Επιπλέον, υπάρχει και ο φάκελος `QUERIES.sql`, που περιέχει όλα τα απαραίτητα `queries` για την υλοποίηση των ζητούμενων λειτουργιών της εφαρμογής.

Η εισαγωγή των `dummy data` γίνεται σε όλα τα στοιχεία, εκτός των βιβλίων, με την βοήθεια της βιβλιοθήκης `faker` την `Python`. Τα ψεύτικα δεδομένα για τα βιβλία έγιναν `generated Google books APIs`.

## QUERIES

Τα παρακάτω `queries` βρίσκονται στον φάκελο `QUERIES.sql`

Queries για Administrator:

### Query 3.1.1

Παρουσίαση λίστας με συνολικό αριθμό δανεισμών ανά σχολείο (Κριτήρια αναζήτησης: έτος, ημερολογιακός μήνας πχ Ιανουάριος).

```
-- 3.1.1
SELECT s.School_Name , Count(B.BorrowID) AS TotalLoans
FROM School_Unit s
JOIN SchoolUser su ON s.SchoolID = su.SchoolID
JOIN Borrow b ON su.SchoolUserID = b.SchoolUserID
WHERE YEAR(b.Borrow_Date) = ? AND MONTH(b.Borrow_Date) = ?      -- The "?" will be specified in the application
GROUP BY s.School_Name;
```

### Query 3.1.2

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

```
-- 3.1.2
SELECT a.FullName AS AuthorName, CONCAT(u.FirstName, ' ', u.LastName) AS TeacherName
FROM Category c
JOIN BookCategory bc ON c.CategoryID = bc.CategoryID
JOIN Book b ON bc.BookID = b.BookID
JOIN BookAuthor ba ON b.BookID = ba.BookID
JOIN Author a ON ba.AuthorID = a.AuthorID
JOIN Borrow br ON b.BookID = br.BookID
JOIN SchoolUser su ON br.SchoolUserID = su.SchoolUserID
JOIN AppUser u ON su.SchoolUserID = u.UserID
WHERE c.CategoryName = ? AND su.Position = 'Teacher' AND YEAR(br.Borrow_Date) = YEAR(CURDATE()) - 1;
```

### Query 3.1.3

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

```
-- 3.1.3
SELECT CONCAT(u.FirstName, ' ', u.LastName) AS TeacherName, COUNT(DISTINCT b.BookID) AS NumberOfBooks
FROM SchoolUser su
JOIN AppUser u ON su.SchoolUserID = u.UserID
JOIN Borrow bo ON su.SchoolUserID = bo.SchoolUserID
JOIN Book b ON bo.BookID = b.BookID
WHERE su.Position = 'Teacher' AND YEAR(CURDATE()) - YEAR(u.BirthDate) < 40
GROUP BY su.SchoolUserID
ORDER BY NumberOfBooks DESC;
```

### Query 3.1.4

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



```
-- 3.1.4
CREATE VIEW UnborrowedAuthors AS
SELECT a.FullName AS AuthorName
FROM Author a
LEFT JOIN BookAuthor ba ON a.AuthorID = ba.AuthorID
LEFT JOIN Book b ON ba.BookID = b.BookID
LEFT JOIN Borrow br ON b.BookID = br.BookID
WHERE br.BorrowID IS NULL;
```

### Query 3.1.5

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

```
-- 3.1.5
SELECT o.OperatorID, CONCAT(u.FirstName, ' ', u.LastName) AS OperatorName, COUNT(b.BorrowID) AS NumberOfLoans
FROM Operator o
JOIN AppUser u ON o.OperatorID = u.UserID
JOIN Borrow b ON o.OperatorID = b.OperatorID
WHERE YEAR(b.Borrow_Date) = ?
GROUP BY o.OperatorID
HAVING NumberOfLoans > 20;
```

### Query 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.

```
-- 3.1.6
SELECT bc1.CategoryID AS Category1, bc2.CategoryID AS Category2, COUNT(*) AS Frequency
FROM BookCategory bc1
JOIN BookCategory bc2 ON bc1.BookID = bc2.BookID AND bc1.CategoryID < bc2.CategoryID
JOIN Borrow b ON bc1.BookID = b.BookID
GROUP BY bc1.CategoryID, bc2.CategoryID
ORDER BY Frequency DESC
LIMIT 3;
```

### Query 3.1.7

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

```

-- 3.1.7
WITH AuthorCounts AS (
    SELECT ba.AuthorID, a.FullName, COUNT(*) AS BookCount
    FROM BookAuthor ba
    JOIN Author a ON ba.AuthorID = a.AuthorID
    GROUP BY ba.AuthorID, A.FullName
),
MaxAuthor AS (
    SELECT MAX(BookCount) AS MaxBooks
    FROM AuthorCounts
)
SELECT ac.AuthorID, ac.FullName, ac.BookCount
FROM AuthorCounts ac, MaxAuthor ma
WHERE ac.BookCount >= ma.MaxBooks - 5;

```

### Queries για Operator:

#### Query 3.2.1

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

```

-- 3.2.1
SELECT B.Title, A.FullName
FROM Book B
JOIN BookAuthor BA ON B.BookID = BA.BookID
JOIN Author A ON BA.AuthorID = A.AuthorID
JOIN BookCategory BC ON B.BookID = BC.BookID
JOIN Category C ON BC.CategoryID = C.CategoryID
WHERE B.Title LIKE ? AND A.FullName LIKE ? AND B.SchoolID = ? AND C.CategoryName LIKE ? AND EXISTS (
    SELECT 1 FROM BookCopy BC WHERE BC.BookID = B.BookID
);

```

#### Query 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 Καθυστέρησης).

```
-- 3.2.2
SELECT su.FirstName, su.LastName, b.BookID, DATEDIFF(CURDATE(), bo.Due_Date) AS Delay_Days
FROM Borrow bo
JOIN SchoolUser su ON bo.SchoolUserID = su.SchoolUserID
JOIN Book b ON bo.BookID = b.BookID
WHERE su.FirstName LIKE ?
AND su.LastName LIKE ?
AND bo.Borrow_Status = 'Overdue'
AND DATEDIFF(CURDATE(), bo.Due_Date) > ?
;
```

### Query 3.2.3

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

```
-- 3.3.3
SELECT SU.FirstName, SU.LastName, C.CategoryName, AVG(R.Likert) as AverageRating
FROM Review R
JOIN SchoolUser SU ON R.SchoolUserID = SU.SchoolUserID
JOIN Book B ON R.BookID = B.BookID
JOIN BookCategory BC ON B.BookID = BC.BookID
JOIN Category C ON BC.CategoryID = C.CategoryID
JOIN Operator O ON R.OperatorID = O.OperatorID
WHERE SU.FirstName LIKE ?
AND SU.LastName LIKE ?
AND C.CategoryName LIKE ?
AND O.OperatorID = ?
GROUP BY SU.FirstName, SU.LastName, C.CategoryName;
```

## Queries για App User:

### Query 3.3.1

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

```
-- 3.3.1
SELECT B.Title, A.FullName AS Author, C.CategoryName
FROM Book B
JOIN BookAuthor BA ON B.BookID = BA.BookID
JOIN Author A ON BA.AuthorID = A.AuthorID
JOIN BookCategory BC ON B.BookID = BC.BookID
JOIN Category C ON BC.CategoryID = C.CategoryID
WHERE B.Title LIKE ?
AND A.FullName LIKE ?
AND C.CategoryName LIKE ?
;
```

### Query 3.3.2

Λίστα όλων των βιβλίων που έχει δανειστεί ο συγκεκριμένος χρήστης.

```
-- 3.3.2
SELECT B.Title, Bo.Borrow_Date, Bo.Due_Date
FROM Borrow Bo
JOIN Book B ON Bo.BookID = B.BookID
WHERE Bo.SchoolUserID = ?
;
```

## 1.3. User Manual

**Δίκτυο Σχολικών Βιβλιοθηκών**

[Login](#) [Register](#) Οδηγός Χρήστη

Εγγραφή αν δεν είστε ήδη χρήστης

Είσοδος με τα στοιχεία σας

**Δίκτυο Σχολικών Βιβλιοθηκών**

**Είσοδος**

Όνομα Χρήστη (username):

Κωδικός (password):

[Ξέχασε τον κωδικό](#) [Νέος χρήστης](#) [Κάνε εγγραφή](#)

[Login](#)

Πατήστε για να επιστρέψετε σε αυτή τη σελίδα (αρχική)

Εισάγετε όλα τα στοιχεία που σας ζητούνται!

Αν έχετε ήδη λογαριασμό, συνδεθείτε είτε από το σύνδεσμο κάτω, είτε πατώντας επιστροφή στην αρχική και Login.

**Δίκτυο Σχολικών Βιβλιοθηκών**

**Εγγραφή Χρήστη - Registration**

Στοιχεία Χρήστη:

Όνομα Χρήστη (username):

Όνομα:

Επίθετο:

Ημερομηνία Γέννησης:

E-mail:

Τηλέφωνο:

Κωδικός:

Σχολική Μονάδα:

Κατηγορία Χρήστη:

- ☐ Υπεύθυνος Βιβλιοθήκης
- ☐ Εκπαιδευτικός
- ☐ Μαθητής

[Submit](#)

### Δίκτυο Σχολικών Βιβλιοθηκών

Αλλαγή Κωδικού

Username

Email

**Request Password Reset**

Αν ξεχάσατε τον κωδικό σας, πατήστε «Ξέχασα τον κωδικό μου στο παράθυρο της εισόδου και εισάγετε τα στοιχεία σας εδώ!

Πατώντας “Request Password Reset” θα οδηγηθείτε στην εισαγωγή νέου κωδικού.

## ADMIN VIEW:

Σαν διαχειριστής έχετε δικαίωμα για δημιουργία αντιγράφου και επαναφορά του συστήματος. Πατήστε εδώ για ολικές ενέργειες και εδώ για ανάρτηση ενός αρχείου.

### Δίκτυο Σχολικών Βιβλιοθηκών

[Λίστα Βιβλιοθηκών](#) [Ιστορικό Δανεισμών](#) [Log Out](#)

#### Επιλογές Κεντρικού Χειριστή (Admin)

[Καταχώρηση Βιβλιοθηκών](#)

[Έγκριση Αιτήσεων Εγγραφής Υπεύθυνων Χειριστών](#)

[Δημιουργία Αντίγραφου Ασφάλειας \(Backup\)](#)

Επιλογή αρχείου για επαναφορά:  No file chosen

**Προσοχή!** Απαραίτητο για backup/restore ένα directory:  
C:/xampp/Data\_Bases/MyDatabase/Backup

Έξοδος από το λογαριασμό.

Εισάγετε νέα βιβλιοθήκη πατώντας εδώ.

Για να δείτε ως διαχειριστής όλες τις βιβλιοθήκες.

Για να εγκρίνετε αιτήματα υπεύθυνων χειριστών βιβλιοθηκών. Θα εμφανιστεί η λίστα:

### Δίκτυο Σχολικών Βιβλιοθηκών

Βιβλιοθήκη που προστέθηκε:

Όνομα	Επώνυμο	Email	Τηλέφωνο	Username	Σχολική Μονάδα	Ενέργειες
Σχολική Βιβλιοθήκη	Καταχώρηση	admin@school.gr	210 1234 5678	admin	Σχολική Μονάδα	<a href="#">Εγκρίνω</a>

### Εκκρεμείς Αιτήσεις Διαχειριστών Σχολικών Μονάδων

The school unit was added successfully

Όνομα	Επώνυμο	Email	Τηλέφωνο	Username	Σχολική Μονάδα	Ενέργειες

**Δίκτυο Σχολικών Βιβλιοθηκών**

[Λίστα Βιβλιοθηκών](#)
[Ιστορικό Δανεισμών](#)
[Οδηγός Χρήστη](#)
[Log Out](#)

**Επιλογές Κεντρικού Χειριστή (Admin)**

[Καταχώρηση Βιβλιοθηκών](#)
[Έγκριση Αιτήσεων Εγγραφής Υπεύθυνων Χειριστών](#)
[Δημιουργία Αντίγραφου Ασφάλειας \(Backup\)](#)
[Επαναφορά Συστήματος \(Restore\)](#)

**Ιστορικό Δανεισμών ανά Σχολική Μονάδα**

Χρονιά: 
Μήνας:

Το ιστορικό δανεισμών θα σας κάνει refresh τη σελίδα, δίνοντάς σας την επιλογή να δείτε το ιστορικό δανεισμών (πλήθος δανεισμών για κάποιο μήνα) ανά σχολική μονάδα. Αν υπάρχουν, θα εμφανιστεί το όνομα της σχολικής μονάδας και ο αριθμός δανεισμών για τον μήνα από δίπλα.

**Δίκτυο Σχολικών Βιβλιοθηκών**

**Καταχώρηση Βιβλιοθήκης**

**Απαραίτητη Ενέργεια!**

Πριν την καταχώρηση της βιβλιοθήκης στο σύστημα Δικτύου Σχολικών Βιβλιοθηκών, θα πρέπει να ενεργοποιήσετε τον λογαριασμό της σχολικής μονάδας σας στο Πανελλήνιο Σχολικό Δίκτυο (ΠΣΔ). Κατεβάστε το έντυπο αίτησης, το οποίο πρέπει να αποστείλετε με e-mail σύμφωνα με τις οδηγίες στη σελίδα [helpdesk](#) του ΠΣΔ.

[Έντυπο Αίτησης \[PDF\]](#)

**Εισάγετε τα παρακάτω στοιχεία:**

Ονομασία Σχολείου:

Διεύθυνση:

Πόλη:

Τηλέφωνο:

E-mail:

Ονοματεπώνυμο Δ/τη Σχολείου:

Ονοματεπώνυμο Υπεύθυνου Χειριστή Σχολικής Βιβλιοθήκης:

Κατά την εγγραφή σχολικής μονάδας, βεβαιωθείτε ότι έχει ενεργοποιηθεί ο λογαριασμός της βιβλιοθήκης στο ΠΣΔ. Στη συνέχεια, εισάγετε όλα τα στοιχεία.

## Δίκτυο Σχολικών Βιβλιοθηκών

### Εισαγωγή Βιβλίου

Τίτλος:	<input type="text"/>
Εκδότης:	<input type="text"/>
ISBN:	<input type="text"/>
Authors:	<input type="text"/>
	<input type="button" value="+"/> <input type="button" value="-"/>
Genres:	<div>action and adventure art children drama</div> <small>Hold Ctrl to select multiple genres</small>
New Genre:	<input type="button" value="+"/> <input type="button" value="-"/> <input type="text"/>
Αριθμός Σελίδων:	<input type="text"/>
Περιγραφή:	<input type="text"/>
Εικόνα:	<input type="text"/>
Γλώσσα:	<input type="text" value="-"/>
Keywords:	<input type="text"/>
	<input type="button" value="+"/> <input type="button" value="-"/>
Διαθέσιμα Αντίτυπα:	<input type="text"/>
<input type="button" value="Εισαγωγή Βιβλίου"/>	

Για να εισάγετε νέο βιβλίο στη βιβλιοθήκη σας, συμπληρώστε τη φόρμα με όλα τα στοιχεία.

Πατώντας στις ενδείξεις **+** **||** **-** μπορείτε να προσθέσετε κάποιο συγγραφέα, είδος ή λέξη κλειδί καθώς και να αφαιρέσετε την προσθήκη αν κάνετε λάθος. Τουλάχιστον μία εισαγωγή απαιτείται.

### Λίστα Βιβλίων

Αναζήτηση βιβλίων...

**Τίτλος:** Tracer  
**ISBN:** 9781800324060  
**Εκδότης:** Canelo  
**Συγγραφείς:** Jason Dean  
**Κατηγορίες:** thriller  
**Λέξεις-κλειδιά:** thriller, author, want, korso, tracer, man, unique  
**Διαθέσιμα Αντίτυπα:** 14



**Τίτλος:** The Fall  
**ISBN:** 9781838931773  
**Εκδότης:** Bloomsbury Publishing  
**Συγγραφείς:** Rachael Blok  
**Κατηγορίες:** thriller  
**Λέξεις-κλειδιά:** thriller, jump, atmospheric, secret, woman, fall, family  
**Διαθέσιμα Αντίτυπα:** 6



Οι επιλογές επεξεργασίας και αφαίρεσης βιβλίου είναι δυνατές μόνο για τον υπεύθυνο χειριστή.

[illegible]

Μπορείτε να κάνετε  
στοχευμένη αναζήτηση  
βάσει κατάστασης  
κράτησης και χρονικής  
περιόδου.

Μπορείτε να κάνετε  
στοχευμένη αναζήτηση  
βάσει κατάστασης  
κράτησης και χρονικής  
περιόδου.

Εδώ δεν εμφανίζονται οι εκπρόθεσμοι δανεισμοί.

[illegible]



Μπορείτε να αποδεχτείτε ή να απορρίψετε την εγγραφή νέων χρηστών.

Εκκρεμείς Αιτήσεις Χρηστών Σχολικής Μονάδας					
Όνομα	Επώνυμο	Email	Username	Ρόλος	Ενέργειες
					<button>Εγκριση</button> <button>Απόρριψη</button>
					<button>Εγκριση</button> <button>Απόρριψη</button>
					<button>Εγκριση</button> <button>Απόρριψη</button>
					<button>Εγκριση</button> <button>Απόρριψη</button>
					<button>Εγκριση</button> <button>Απόρριψη</button>
					<button>Εγκριση</button> <button>Απόρριψη</button>
					<button>Εγκριση</button> <button>Απόρριψη</button>
					<button>Εγκριση</button> <button>Απόρριψη</button>
					<button>Εγκριση</button> <button>Απόρριψη</button>
					<button>Εγκριση</button> <button>Απόρριψη</button>
					<button>Εγκριση</button> <button>Απόρριψη</button>
					<button>Εγκριση</button> <button>Απόρριψη</button>

## ALL USER VIEW:

### Δίκτυο Σχολικών Βιβλιοθηκών

[Ιστορικό Δανεισμών](#)

Οδηγός Χρήστη

[Log Out](#)

Δείτε προηγούμενους δανεισμούς σας.

## 1.4.

### Οδηγίες Εγκατάστασης

Βήμα 1<sup>ο</sup>: Κατεβάστε το repository μέσω Github. Αυτό μπορείτε να το κάνετε πηγαίνοντας στο σύνδεσμο της παραγράφου 1.5 και κατεβάζοντας τα αρχεία που βρίσκονται στο φάκελο main.

Βήμα 2<sup>ο</sup>: Για να εγκαταστήσετε τη βάση και να εισάγετε τα δεδομένα σε αυτή, μπορείτε να χρησιμοποιήσετε SQL server μέσω Xampp (ξεκινώντας την Mysql και τον Apache) και στη συνέχεια το Mysql Workbench, όπου θα τρέξετε τα αρχεία DDL\_SCRIPT.sql και TRIGGERS\_EVENTS\_PROCEDURES.sql μαζί με τα αρχεία δεδομένων.

Βήμα 3<sup>ο</sup>: Κάνετε εγκατάσταση τις απαραίτητες βιβλιοθήκες μέσω των εντολών:

```
pip install flask
```

```
pip install flask-wtf
```

```
pip install flask_sqlalchemy
```

Για τη δημιουργία του Application χρησιμοποιήθηκε PYCHARM, όπου φτιάχνουμε νέο Project στο οποίο βάζουμε το πακέτο της εφαρμογής μας. Στο Terminal του Project εισάγουμε τις παραπάνω εντολές και αφού γίνουν οι εγκαταστάσεις, τότε τρέχουμε το αρχείο run.py.

## 1.5. Σύνδεσμος του Git Repo

Το repository της βάσης δεδομένων στο Github είναι:

<https://github.com/TilemachosNots21/School-Library-Management>