

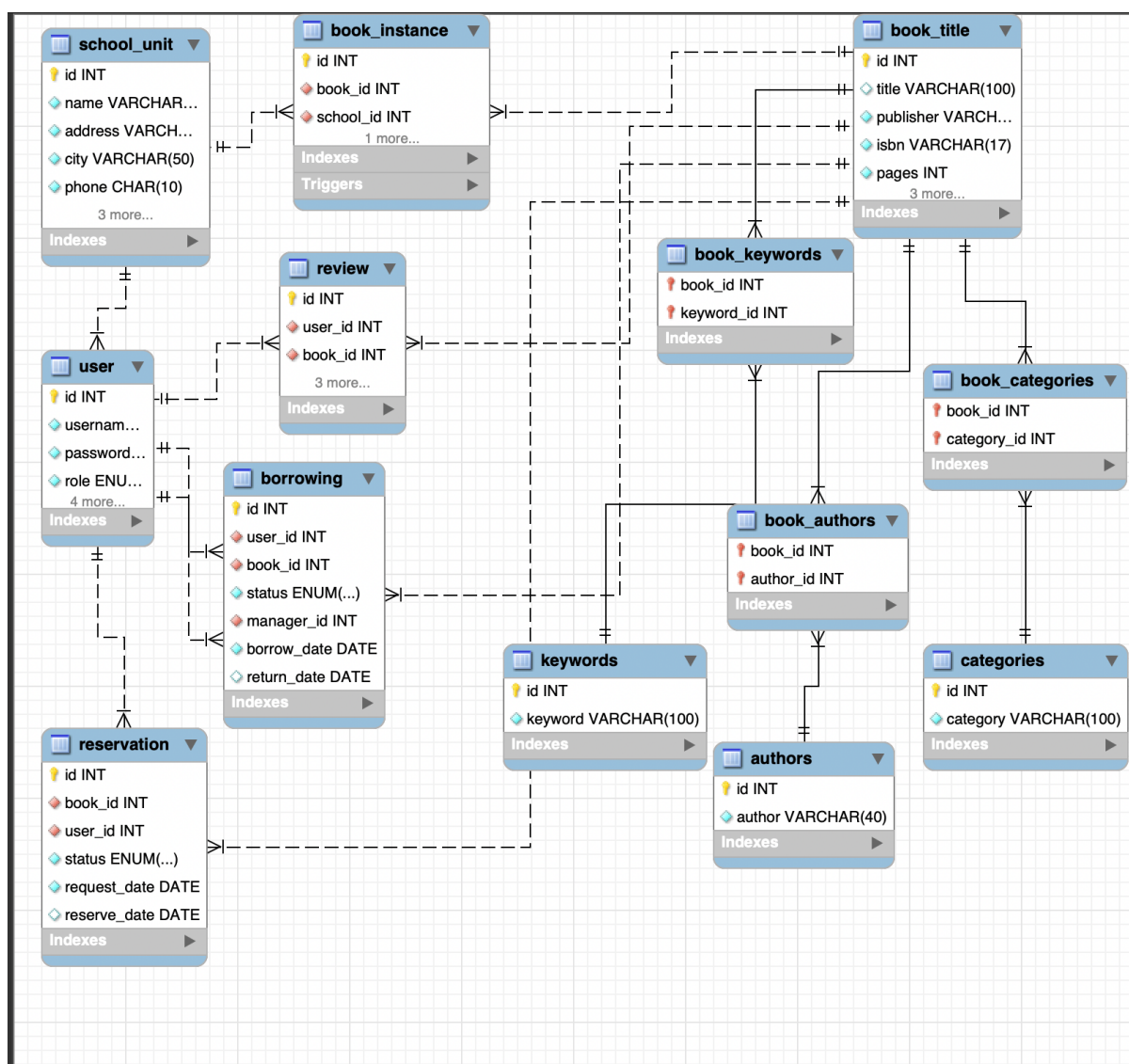
## ΑΝΑΦΟΡΑ ΕΞΑΜΗΝΙΑΙΑΣ ΕΡΓΑΣΙΑΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

ΣΠΟΥΔΑΣΤΕΣ: ΣΤΕΦΑΝΟΣ ΓΙΑΝΝΑΚΟΠΟΥΛΟΣ (03120829) και ΕΥΑΓΓΕΛΟΣ ΠΙΠΗΣ (03120630)

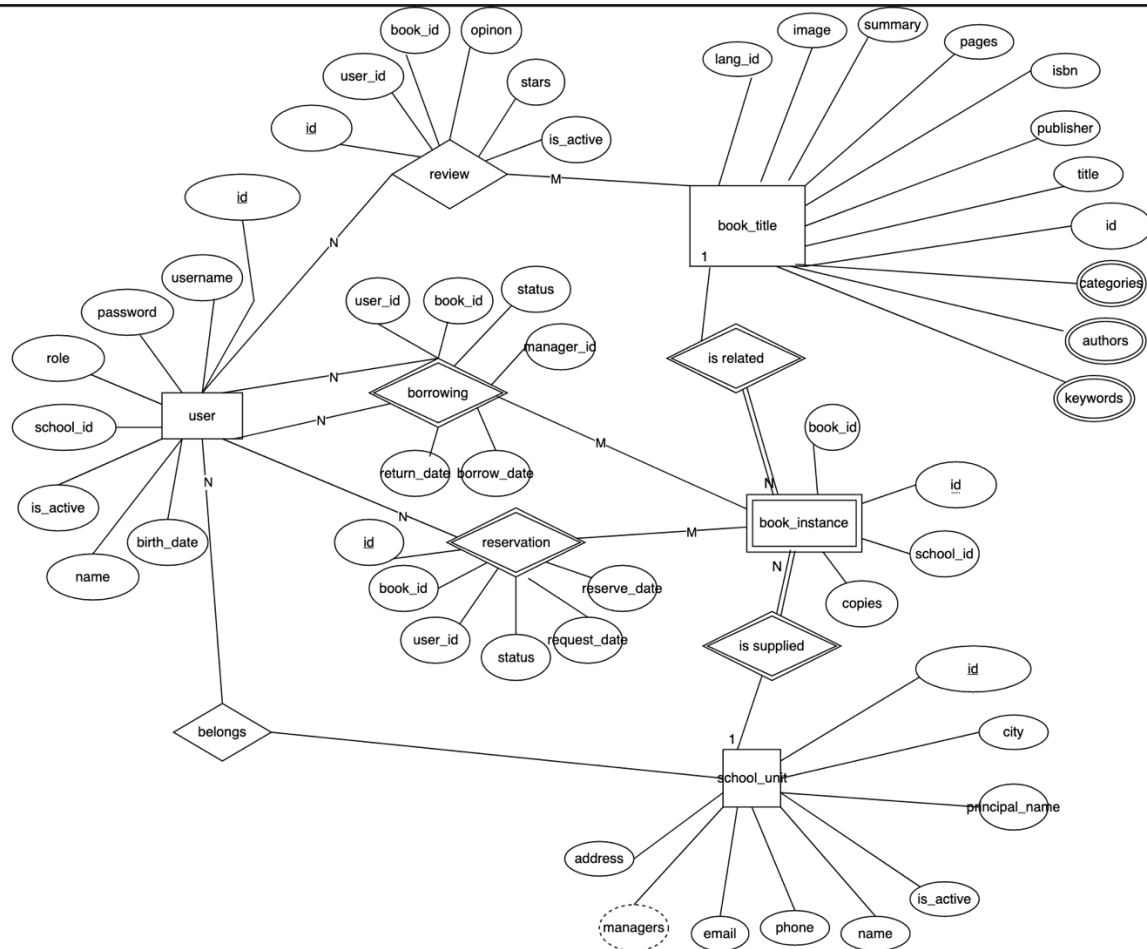
ΣΠΟΥΔΑΣΤΡΙΑ: ΜΑΡΙΑ ΑΡΓΥΡΩ ΛΑΖΟΥ (03120129)

ΚΟΜΜΑΤΙ: 1.1)

Σχεσιακό Διάγραμμα:



ER:



#### Ανάλυση των πινάκων για ER και σχεσιακό:

1. **school\_unit**: Στον πίνακα αυτόν αποθηκεύουμε τις πληροφορίες για κάθε σχολική μονάδα. Τα πεδία που αποθηκεύονται είναι το id του σχολείου (row identifier), το όνομα του σχολείου, η διεύθυνση, η πόλη, ο αριθμός τηλεφώνου, το email, το ονοματεπώνυμο του διευθυντή, καθώς και ένα πεδίο τύπου bool με το οποίο μπορούμε να αποφανθούμε για το εάν η σχολική μας μονάδα είναι εγκεκριμένη από τον κεντρικό διαχειριστή. Ως primary key έχουμε το id του σχολείου.
2. **user**: Ο πίνακας του χρήστη αποτελείται από τα πεδία: ονοματεπώνυμο, κωδικός, ρόλος που είναι ένα πεδίο τύπου enum και περιλαμβάνει τους 4 πιθανούς ρόλους ενός χρήστη στην εφαρμογή, δηλαδή καθηγητής, μαθητής, κεντρικός διαχειριστής και διαχειριστής σχολικής βιβλιοθήκης. Ακόμα, έχουμε πεδίο για σύνδεση του χρήστη μας με το σχολείο στο οποίο ανήκει, το οποίο το δηλώνουμε μέσω του πεδίου school\_id, το πεδίο αυτό είναι ουσιαστικά ένα foreign key που δείχνει στο πεδίο id του πίνακα school\_unit και δηλώνει την σχέση 1-many μεταξύ σχολικής μονάδας και user (δηλαδή ότι μια σχολική μονάδα έχει και πολλούς χρήστες), ενώ ταυτόχρονα επιτυγχάνεται σωστά το mapping του user με το σχολείο στο οποίο ανήκει. Τέλος,

έχουμε τα πεδία `is_active`, `name` και `birth_date`. Με το πρώτο μπορούμε να αποφανθούμε για το εάν ο χρήστης μας έχει γίνει αποδεκτός από τον αντίστοιχο διαχειριστή ή και διαχειριστές της βιβλιοθήκης του σχολείου του, με το 2<sup>ο</sup> προσθέτουμε στην βάση το ονοματεπώνυμο του χρήστη και με το 3<sup>ο</sup> την ακριβή ημερομηνία γέννησης του. Ως primary key έχουμε το `id` του χρήστη.

3. `book_title`: Ο συγκεκριμένος πίνακας χρησιμοποιείται, ώστε να αποθηκεύονται τα βιβλία όχι ως φυσικά αντίτυπα, αλλά ως ένα αφηρημένο σύνολο το οποίο περιγράφει τα γενικά χαρακτηριστικά του βιβλίου. Αποτελείται από τα πεδία: `title`, `publisher`, `isbn`, `pages`, `summary`, `image` και `lang_id`, τα οποία είναι ατομικά. Στους αντίστοιχους πίνακες γίνεται καταχώρηση: του τίτλου (το πεδίο αυτό το επιλέξαμε να είναι `unique` ώστε να μην καταχωρείται περισσότερες από 1 φορά το ίδιο βιβλίο\*\*), του εκδότη, το `isbn`, ο αριθμός των σελίδων, η περίληψη της υπόθεσης, μια εικόνα η οποία αποθηκεύτηκε με την μορφή `url` (ώστε να μην γεμίζει η βάση χωρίς λόγο) και τέλος ένα `id` 2 χαρακτήρων που δηλώνει σε ποια γλώσσα είναι γραμμένο το βιβλίο (πχ `en`->αγγλικά και `gr`->Ελληνικά). Ως primary key έχουμε το πεδίο `id`.

*\*\*σχόλιο: Το `isbn` ίσως είναι πιο εύλογη επιλογή, όμως κάνουμε την παραδοχή ότι ένας τίτλος δεν εμφανίζεται σε άλλο βιβλίο με διαφορετικούς εκδότες ή συγγραφείς για ευκολία δοκιμών της εφαρμογής μας όταν καταχωρούνται δανεισμοί, κρατήσεις ή γίνεται προσθήκη νέου βιβλίου.*

***Τα επόμενα πεδία αποτελούν τα multivalued attributes του `book_title` και επειδή έχουμε many to many relationships καταχωρούνται αναγκαστικά σε ξεχωριστούς πίνακες.***

Συγκεκριμένα:

4. `categories`: Ο πίνακας αυτός χρειάστηκε επειδή ένα βιβλίο μπορεί να ανήκει σε περισσότερες από μια κατηγορία. Αποτελείται από μόλις 2 πεδία, το `id` καθώς και το πεδίο `category`, που επιλέξαμε να είναι `unique`, ώστε να μην καταχωρείται η ίδια κατηγορία περισσότερες από 1 φορά. Ως primary key έχουμε το πεδίο `id`.
5. `book_categories`: Ο πίνακας αυτός αποτελεί ουσιαστικά το μέσο για να κάνουμε σωστά το mapping του κάθε βιβλίου με τις κατηγορίες του. Μέσω αυτού του πίνακα (του λεγόμενου ενδιάμεσου), κατορθώνουμε πέραν της ορθής τοποθέτησης των κατηγοριών να δηλώσουμε και την many to many σχέση που υπάρχει μεταξύ μιας κατηγορίας και ενός βιβλίου. Η many to many αυτή σχέση δηλώνεται μέσω 2 foreign keys του πίνακα μας, που αποτελούν ουσιαστικά και τα μοναδικά πεδία του: `book_id` and `category_id`, τα πεδία αυτά όπως μαρτυρά και το όνομα δείχνουν στο `id` του

book\_title καθώς και στο id του πίνακα categories αντίστοιχα. Ως primary key, έχουμε τον συνδυασμό των 2 πεδίων αυτών και με τον τρόπο αυτό, διασφαλίζουμε να μην αντιστοιχίσουμε ένα βιβλίο σε μια συγκεκριμένη κατηγορία περισσότερες από 1 φορά και το αντίστροφο.

6. authors: Όπως και στην περίπτωση του πίνακα categories, ο πίνακας αυτός χρειάστηκε επειδή ακριβώς ένα βιβλίο μπορεί να έχει περισσότερους από έναν συγγραφείς. Αποτελείται από 2 πεδία το id καθώς και το πεδίο author, το οποίο είναι και unique ώστε να μην καταχωρείται στην βάση μας περισσότερες από 1 φορά ο ίδιος συγγραφέας. Ως primary key έχουμε το id.
7. book\_authors: Μέσω αυτού γίνεται σωστά το mapping του κάθε βιβλίου με τους συγγραφείς του, ενώ παράλληλα δηλώνεται η many to many σχέση μεταξύ των πινάκων (book\_title και authors), καθώς ένας συγγραφέας μπορεί να έχει γράψει πολλά βιβλία και συγχρόνως ένα βιβλίο να έχει πολλούς συγγραφείς. Αποτελείται από 2 πεδία το book\_id και το author\_id, 2 foreign keys που δείχνουν στα id's των book\_title και authors αντίστοιχα. Ως primary key, έχουμε τον συνδυασμό των 2 πεδίων αυτών και με τον τρόπο αυτό, διασφαλίζουμε να μην αντιστοιχίσουμε ένα βιβλίο στον ίδιο συγγραφέα περισσότερες από 1 φορά και το αντίστροφο.
8. keywords: Αποτελείται από 2 πεδία: id και keyword με constraint unique, ώστε να έχουμε εισαγωγή μόνο διαφορετικών keywords. Χρειάζεται καθώς ένα βιβλίο μπορεί να έχει πολλά keywords, αλλά και πολλά keywords να αντιστοιχίζονται σε περισσότερα από 1 βιβλία. Ως primary key έχουμε το πεδίο id.
9. book\_keywords: Αξιοποιείται για να δηλώσουμε την many to many σχέση μεταξύ των πινάκων keywords και book\_title. Αποτελείται από 2 πεδία: book\_id και keyword\_id τα οποία είναι foreign keys και δείχνουν αντίστοιχα στα πεδία id των πινάκων book\_title και keywords αντίστοιχα, μέσω των 2 αυτών κλειδιών επιτυγχάνουμε να δηλώσουμε την many to many σχέση που αναφέρθηκε παραπάνω. Παράλληλα, μέσω αυτού του πίνακα γίνεται σωστά η αντιστοίχιση του κάθε βιβλίου με τις λέξεις κλειδιά του. Ως primary key έχουμε τον συνδυασμό των 2 αυτών πεδίων, με τον τρόπο αυτό διασφαλίζουμε για ένα βιβλίο να μην καταχωρηθεί η ίδια κατηγορία περισσότερες από 1 φορά.
10. book\_instance: Ο πίνακας αυτός αντιπροσωπεύει την φυσική αναπαράσταση ενός βιβλίου σε επίπεδο σχολικής μονάδας. Διαφορετικά σχολεία διαθέτουν ξεχωριστά book\_instances του ίδιου βιβλίου (book\_title) στις βιβλιοθήκες τους (ποικίλλουν στον αριθμό αντιτύπων και είναι ορατά μόνο για τους χρήστες του συγκεκριμένου σχολείου)

όμως περιγράφονται από τα ίδια γενικά χαρακτηριστικά. Ουσιαστικά αποτελεί την ενσάρκωση του `book_title` που είναι άυλο αντικείμενο. Επειδή ένα `book_instance` δεν μπορεί να υπάρξει χωρίς την ύπαρξη του αντίστοιχου τίτλου αναπαρίσταται ως `weak entity set` και συνδέεται με αυτό μέσω της σχέσης `is related`. (Τονίζουμε ότι πληρεί τα κριτήρια του `total participation` και της απουσίας `primary key`). Αντίστοιχα προκύπτουν και οι `weak σχέσεις` : `is supplied`, `borrowing`, `reservation`, καθώς αποτελούν ενέργειες που αδυνατούν να πραγματοποιηθούν χωρίς την φυσική υπόσταση του βιβλίου στο αντίστοιχο σχολείο. Αποτελείται από τα πεδία: `book_id` και `school_id`, τα οποία είναι `foreign keys` και δείχνουν στα πεδία `id` των πινάκων `book_title` και `school_unit` αντίστοιχα, μέσω αυτών επιτυγχάνουμε την δήλωση της `many to many` σχέσης μεταξύ `book_title` και `school_unit` (ένα σχολείο εν τέλει μπορεί να έχει πολλά αντίτυπα, αλλά και ένας συγκεκριμένος τίτλος μπορεί να υπάρχει σε περισσότερα από 1 σχολεία) και από ένα πεδίο `copies` στο οποίο καταχωρούμε το πλήθος των αντιτύπων που έχει μια σχολική μονάδα για τον συγκεκριμένο τίτλο. Ως `weak key attribute` θέτουμε ένα πεδίο `id`, το οποίο σε συνδυασμό με το `book_id` συμπληρώνει την σχέση `is related` και σε συνδυασμό με το `school_id` συμπληρώνει την σχέση `is supplied`.

11. `review`: Ο πίνακας `review` χρησιμοποιείται για να καταχωρούμε στην βάση τα σχόλια του κάθε χρήστη. Ταυτόχρονα δηλώνει την `many to many` σχέση μεταξύ των πινάκων `user` και `book_title` (ένας χρήστης μπορεί να κάνει πολλά `reviews` και ένας τίτλος μπορεί να έχει πολλά `reviews` από διαφορετικούς χρήστες). Σημειώνεται ότι εδώ δεν είναι απαραίτητη η ύπαρξη του φυσικού αντιτύπου σε κάποια σχολική μονάδα αφού από την εκφώνηση ένας χρήστης μπορεί να κάνει κριτική σε οποιοδήποτε βιβλίο χωρίς να το έχει δανειστεί, γι' αυτό δεν συνδέεται με το `book_instance` αλλά απευθείας με το `book_title`. Αποτελείται από τα πεδία: `user_id` και `book_id`, τα οποία είναι και τα `foreign keys` που δείχνουν στα `id` των πινάκων `user` και `book_title` αντίστοιχα, μέσω αυτών δηλώνεται και η παραπάνω `many to many` σχέση και γίνεται σωστά το `mapping` ενός `review` με τον χρήστη που το έκανε. Ακόμα, αποτελείται από ένα πεδίο `opinion` που αποτελεί και ουσιαστικά το `review` ενός χρήστη για τον τίτλο, ένα πεδίο `stars` (`rating`) που δηλώνει τον αριθμό των αστεριών από το 1-5 που μπορεί να βάλει ο χρήστης στο συγκεκριμένο βιβλίο και τέλος το πεδίο `is_active`, που ανάλογα με τον ρόλο του χρήστη μπορεί να καταχωρηθεί από `default` αρχικά `true` ή και `false`, ειδικότερα το `review` ενός μαθητή πρέπει να εγκριθεί από κάποιον διαχειριστή της σχολικής του βιβλιοθήκης πριν ενεργοποιηθεί και μπορούν και οι άλλοι χρήστες να το δουν, στην περίπτωση ενός καθηγητή δεν είναι απαραίτητη η έγκριση και με το που κάνει `submit` το `review` του όλοι οι χρήστες μπορούν και να το δουν. Τέλος έχουμε και ένα πεδίο `id` το οποίο είναι το `primary key` μας.

12. `borrowing`: Μέσω του πίνακα μας γίνεται η καταχώρηση στην βάση μας των δανεισμών των βιβλίων, ενώ ταυτόχρονα δηλώνονται 2 `many to many` σχέσεις. Συγκεκριμένα, η `many to many` σχέση μεταξύ του `user` και του `book_instance` καθώς και η `many to many` σχέση μεταξύ `user` (ως διαχειριστής, μέσω του οποίου επικυρώνονται οι δανεισμοί) και `user` (ως δανειζόμενος). Είναι `weak` για του λόγους που εξηγήσαμε νωρίτερα (δεν πραγματοποιείται δανεισμός χωρίς την ύπαρξη αντιτύπου στο συγκεκριμένο σχολείο). Έχει τα ακόλουθα πεδία: `id`, `user_id` που κάνει

reference ως foreign key στο id του user και δηλώνει πως ουσιαστικά ένας χρήστης μπορεί να κάνει πολλούς δανεισμούς, ένα πεδίο book\_id που κάνει reference ως foreign key στο δανεισμένο βιβλίο που δηλώνει ότι ένα βιβλίο μπορεί να συμμετέχει σε πολλούς δανεισμούς και ο συνδυασμός των 2 αυτών foreign keys εγκαθιδρύει ουσιαστικά την πρώτη many to many σχέση που περιγράφεται από πάνω. Ακόμα έχουμε ως πεδία: ένα status τύπου enum ['active', 'delayed', 'completed'] που δηλώνει την κατάσταση στην οποία βρίσκεται η κράτηση, ένα foreign key manager\_id που δείχνει πάλι στο user\_id και με αυτό ουσιαστικά κατορθώνουμε να εγκαθιδρύσουμε και την 2<sup>η</sup> many to many σχέση που περιγράφεται παραπάνω, τέλος έχουμε ακόμα 2 πεδία, το πρώτο είναι το borrow\_date και το δεύτερο το return\_date. Ως primary key έχουμε το πεδίο id.

13. reservation: Μέσω του πίνακα των κρατήσεων, γίνεται στην βάση μας η καταχώρηση των αιτημάτων για κράτηση, συγχρόνως δηλώνεται η many to many σχέση μεταξύ user και book\_instance, ότι δηλαδή ακριβώς ένας χρήστης μπορεί να κάνει περισσότερες από 1 αιτήσεις για δανεισμό κατά την διάρκεια ύπαρξης του ως user στην εφαρμογή συνολικά, αλλά ταυτόχρονα ότι και ένα βιβλίο μπορεί να κρατηθεί όχι μόνο από έναν αλλά και περισσότερους χρήστες. Ακολουθεί την λογική του borrowing ως προς το weakness και περιλαμβάνει τα παρακάτω πεδία: 2 foreign keys book\_id user\_id, τα οποία δείχνουν στα id's του book\_instance και user\_id αντίστοιχα εγκαθιδρύοντας την αναφερόμενη many to many σχέση, ένα πεδίο status τύπου enum που μας δείχνει σε τι στάδιο βρίσκεται ο κύκλος ζωής του αιτήματος της κράτησης, καθώς και ένα request\_date και ένα reserve\_date. Τέλος, περιλαμβάνει και ένα πεδίο id το οποίο είναι και το foreign key μας.

### ***\*\* Γενικό σχόλιο για το ER \*\* :***

Επιλέξαμε τα strong entities να είναι 3 μόνο για λόγους scalability.

### ***\*\* Σημαντική τροποποίηση στην υλοποίηση \*\* :***

Για λόγους ευκολίας κατά το data fetching μέσα στην εφαρμογή αντιστοιχούμε απευθείας έναν δανεισμό borrowing, ή μια κράτηση reservation με έναν τίτλο βιβλίου book\_title και όχι book\_instance ελέγχοντας manually τους περιορισμούς να υπάρχει ως φυσικό αντίτυπο στο εκάστοτε σχολείο του αιτούντος (γι' αυτό τα foreign keys στους πίνακες μας δείχνουν σε αυτό το πεδίο)

### ***Ευρετήρια (indexes) :***

Λαμβάνοντας υπόψιν τα παρακάτω:

- Αυτόματα δημιουργούνται indexes από την MySQL στα primary keys των tables που έχουμε στην βάση μας αλλά και στις στήλες που έχουμε unique constraints.

- Στην MySQL δεν δημιουργούνται αυτόματα τα indexes στα foreign keys των tables που έχουμε στην βάση μας.
- Στους many to many πίνακες της βάσης μας, όπου έχουμε και σύνθετα primary keys, δημιουργούνται μεν multicolumn indexes αλλά δε αξιοποιούνται αν και μόνο αν η αναζήτηση μας πραγματοποιηθεί αποκλειστικά συναρτήσει του πρώτου πεδίου του «multi column» index μας.
- Τα indexes μας γλιτώνουν χρόνο από την αναζήτηση, ωστόσο προσθέτουν χρόνο στο update των πινάκων μας στα αντίστοιχα πεδία, οπότε πρέπει να τοποθετούνται με προσοχή. Στο project μας θεωρήσαμε ότι είναι αναγκαίο να τοποθετηθούν indexes (πέρα αυτών που δημιουργούνται από default) αποκλειστικά στα πεδία τα οποία λαμβάνουν και πρωταγωνιστικό ρόλο στα διάφορα queries που κληθήκαμε να αναπτύξουμε.

### ***Πιο συγκεκριμένα:***

Τοποθετήσαμε indexes στα παρακάτω πεδία:

Βάζουμε πρώτα 2 indexes στα many to many tables που κάνουν ουσιαστικά mapping του bookTitle ανάλογα με την κατηγορία και τους συγγραφείς μέσω του id του βιβλίου.

```
create index book_title_id_idx on book_categories(book_id);
create index book_title_id_idx on book_authors(book_id);
```

Διασφαλίσαμε στους 2 αυτούς πίνακες ότι η αναζήτηση με όποιον τρόπο και να γίνει θα έχουμε index, άρα θα γίνει σημαντικά πιο γρήγορα αν έχουμε ΠΟΛΛΕΣ εγγραφές.

Θεωρούμε ότι παρόλο που θα ανανεώνονται αυτοί οι πίνακες συχνά, επειδή χρησιμοποιούνται τα πεδία αυτά σε αρκετά queries θα έχουμε σημαντικά περισσότερες αναζητήσεις από ότι εισαγωγές, οπότε κρίνεται και αναγκαία η τοποθέτηση indexes.

Επιθυμούμε το mapping της σχολικής βιβλιοθήκης να γίνεται γρήγορα συναρτήσει του user, ώστε να του εμφανίζονται και τα κατάλληλα αποτελέσματα βάσει της σχολικής του βιβλιοθήκης.

```
create index school_unit_id_idx on user(school_id);
```

Επιθυμούμε το mapping του review να γίνεται γρήγορα τόσο συναρτήσει του user όσο και συναρτήσει του book\_title.

```
create index user_id_idx on review(user_id);
create index book_title_idx on review(book_id);
```

Καταφέραμε το mapping του review να γίνεται σημαντικά πιο γρήγορα σε ΠΟΛΛΕΣ εγγραφές αν η αναζήτηση γίνει είτε με το id του user είτε με το id του τίτλου για το οποίο έγινε το review.

Αντίστοιχα στον πίνακα book\_keywords όπου γίνεται το mapping βιβλίου και keyword είναι αναγκαίο να βάλουμε index στο πεδίο book\_id, αυτό γιατί αν και έχουμε 2πλο primary key άρα multicolumn index, γνωρίζουμε ότι στην MySQL αν κάνουμε search με βάσει ένα τέτοιο index δεν θα επωφεληθούμε απαραίτητα, διότι το indexing γίνεται συναρτήσει του πρώτου όρου του διπλού κλειδιού, οπότε αν θελήσουμε να κάνουμε search συναρτήσει του 2ου δεν θα επωφεληθούμε από το index, πρέπει να βάλουμε index και στον 2ο όρο.

```
create index book_title_id_idx on book_keywords(book_id);
```

Ανάλογα, παρατηρούμε ότι χρησιμοποιείται στα queries μας και το πεδίο copies του πίνακα book\_instance, πρέπει να βάλουμε index και στο πεδίο αυτό για τους λόγους που αναφέρθηκαν παραπάνω.

```
create index number_of_copies_idx on book_instance(copies);
```

Ομοίως, τοποθετούμε ένα index και στο πεδίο borrow\_date του πίνακα borrowing  

```
create index borrowing_date_idx on borrowing(borrow_date).
```

Τέλος, επιλέγουμε να βάλουμε index και στο πεδίο birth\_date του πίνακα user, καθώς και αυτό χρησιμοποιείται σε ένα query.

```
create index date_of_birth_idx on user(birth_date);
```

Συνολικά προσθέσαμε 9 indexes στα ήδη υπάρχοντα(λόγω των primary keys και των unique constraints). Για αυτόν ακριβώς το λόγο δεν προσθέσαμε indexes και στα πεδία author αλλά και category, επειδή ακριβώς είναι unique και η MySQL δημιουργεί αυτόματα index.

## 1.2

DDL και DML script επισυνάπτονται στην υποβολή.

## 1.4

Περιγράφεται στο Github:

[https://github.com/evpipis/school\\_library/blob/main/README.md](https://github.com/evpipis/school_library/blob/main/README.md)



1.5

Σύνδεσμος για το git repo: [https://github.com/evpipis/school\\_library](https://github.com/evpipis/school_library)