

# Ομάδα Project 1 - School-Library-App DB Report

Παναγιώτης Κωνσταντίνος Κακκαβάς

Αχιλλεύς Άγγελος Μπραϊμάκης

Νικόλαος Παπακωνσταντόπουλος

## Entity Relationship Diagram

Υπάρχουν τρεις οντότητες, καθεμιά από τις οποίες έχει ως primary key το υπογραμμισμένο χαρακτηριστικό της (**User** -> username, **School\_Library** -> address, **Book** -> ISBN)

Το πεδίο type του User μπορεί να πάρει μόνο τις εξής τιμές: admin, librarian, teacher, student.

Τα χαρακτηριστικά authors, topic, keywords είναι multivariable και για αυτό στο σχεσιακό διάγραμμα θα χρησιμοποιηθεί ξεχωριστός πίνακας για καθένα από αυτά συνδέοντάς τα με την κύρια οντότητα το βιβλίο.

Τα χαρακτηριστικά number\_of\_borrowed\_books, number\_of\_reserved\_books, delayed\_and\_not\_returned είναι derived και χρησιμοποιούνται στα triggers για την εισαγωγή στους πίνακες Borrowing, Reservation (έχουν φτιαχτεί κατάλληλα views για να διευκολύνουν τα ερωτήματα).

Για το relationship **Signup\_Approval**:

Όλοι οι χρήστες (total participation) εγκρίνονται από ένα ακριβώς σχολείο και ένα σχολείο μπορεί να έχει πολλούς μαθητές (1-N relationship) (αλλά μπορεί να μην έχει και κανέναν τη στιγμή που εισάγεται στο σύστημα...)

Για το relationship **Available**:

Όλα τα βιβλία (total participation) που υπάρχουν στο σύστημα υπάρχουν σε τουλάχιστον ένα σχολείο. Για συγκεκριμένο σχολείο και βιβλίο που ανήκει στο σχολείο, υπάρχει ο αριθμός αντιτύπων books\_number του βιβλίου στο σχολείο αυτό (μπορεί να γίνει και μηδέν όταν εξαντληθούν όλα τα αντίτυπα). Ένα σχολείο μπορεί να έχει πολλά βιβλία (αλλά μπορεί να μην έχει και κανένα τη στιγμή που εισάγεται στο σύστημα...) και ένα βιβλίο μπορεί να υπάρχει σε πολλά σχολεία (αφού αντιμετωπίζουμε το βιβλίο ως ISBN) (M-N relationship)

Για το Relationship **Review**:

Όλοι οι χρήστες μπορούν να βαθμολογήσουν από κανένα έως πολλά βιβλία, και ένα βιβλίο μπορεί να έχει από καμία ως πολλές κριτικές (M-N relationship)

Για το Relationship **Reservation**:

Όλοι οι χρήστες μπορούν να κάνουν κράτηση από κανένα έως 2 ανά εβδομάδα βιβλία από το σχολείο τους. Ο περιορισμός των δύο κρατήσεων δεν φαίνεται στο ER αλλά υλοποιείται στο trigger για την εισαγωγή στον πίνακα Reservation. Ένα βιβλίο

σε ένα σχολείο μπορεί να έχει “κρατηθεί” από κανέναν έως πολλούς χρήστες. Επίσης, σε ένα σχολείο μπορεί να έχουν γίνει καμία έως πολλές κρατήσεις.

Για το Relationship **Borrowing**:

Όλοι οι χρήστες μπορούν να δανειστούν από κανένα έως 2 ανά εβδομάδα βιβλία από το σχολείο τους. Ο περιορισμός των δύο δανεισμών δεν φαίνεται στο ER αλλά υλοποιείται στο trigger για την εισαγωγή στον πίνακα Borrowing. Ένα βιβλίο σε ένα σχολείο μπορεί να έχει δανειστεί από κανέναν έως πολλούς χρήστες (προφανώς για διαφορετική χρονική περίοδο). Επίσης, σε ένα σχολείο μπορεί να έχουν γίνει κανένας έως πολλοί δανεισμοί. Ο User συμμετέχει με δύο ρόλους στην σχέση του δανεισμού: ως borrower και ως librarian που ενέκρινε τον δανεισμό

Για τα Relationships **Borrowing & Reservation**:

Τα end\_date() και για τους δύο πίνακες υπολογίζονται ως 7 ημέρες μετά την start\_date και χρησιμοποιούνται στα views, triggers και procedures για εύρεση καθυστερημένων δανεισμένων βιβλίων και για διαγραφή κρατήσεων που έγιναν πριν από μια βδομάδα.

### **Relational Schema**

Για τα χαρακτηριστικά authors, topic, keywords που είναι multivariable στο σχεσιακό διάγραμμα έχει χρησιμοποιηθεί ξεχωριστός πίνακας για καθένα από αυτά συνδέοντάς τα με την κύρια οντότητα το βιβλίο μέσω του foreign key ISBN.

Για το relationship **Signup\_Approval**:

Εφόσον είναι 1-N σχέση αρκεί να δοθεί στον User το foreign key address που υποδηλώνει τη διεύθυνση του Σχολείου του.

Για το relationship **Available**:

Είναι M-N relationship οπότε χρησιμοποιούμε ξεχωριστό πίνακα με foreign keys address (School\_Library) και ISBN (Book).

Για το Relationship **Review**:

Είναι M-N relationship οπότε χρησιμοποιούμε ξεχωριστό πίνακα με foreign keys username (User) , ISBN (Book).

Για το Relationship **Reservation**:

Είναι M-N relationship οπότε χρησιμοποιούμε ξεχωριστό πίνακα με foreign keys username (User), address (School\_Library), ISBN (Book).

Για το Relationship **Borrowing**:

Είναι M-N relationship οπότε χρησιμοποιούμε ξεχωριστό πίνακα με foreign keys username (User), address (School\_Library), ISBN (Book), librarian (User).

(\*)Αξίζει να σημειωθεί ότι οι Σχέσεις *Reservation*, *Borrowing* θα μπορούσαν να εξαρτώνται μόνο από τον χρήστη και το βιβλίο δεδομένου ότι ο χρήστης μπορεί να δανειστεί μόνο από το σχολείο του (όπως δηλώνεται στις προδιαγραφές του συστήματος). Η εξάρτηση από το σχολείο θα μπορούσε να εκφραστεί από το *foreign key address* του *User (borrower)*. Ωστόσο η επιλογή του *address* ως *foreign key* βοηθάει στην αποφυγή κάποιων *joins* (π.χ. όταν θέλουμε να διαλέξουμε όλους τους δανεισμούς που γίνονται σε κάποιο σχολείο).

Επίσης, θα μπορούσε το *address* να μην είναι μέρος του *primary key*. Οπότε το ελάχιστο κλειδί για τους αντίστοιχους πίνακες θα ήταν (*username (User)*, *ISBN (Book)*, *start\_date*). Αυτά τα στοιχεία φτάνουν για να προσδιορίσουν μοναδικά κάποιον δανεισμό.

## **Indexes**

Τα ευρετήρια είναι δομές (συνήθως B δέντρα) που βοηθούν στην γρήγορη αναζήτηση σε έναν πίνακα της βάσης δεδομένων και βασίζονται σε ένα χαρακτηριστικό (*column*) του πίνακα. Ο χρόνος αναζήτησης γίνεται συνήθως λογαριθμικός του πλήθους των στοιχείων του πίνακα όταν αναζητούμε με βάση αυτό το στοιχείο ενώ χωρίς το *index* θα ήταν μάλλον γραμμικός...

π.χ. για το "index\_first" το query

```
select * from User where first_name='Nikolaos';
```

γίνεται πιο γρήγορα όταν έχει φτιαχτεί το ευρετήριο *index\_first*

Ωστόσο η εισαγωγή στοιχείων σε πίνακα με ευρετήριο γίνεται σε χρόνο λογαριθμικό (αντί για σταθερό), οπότε η δημιουργία ενός ευρετηρίου αυξάνει τον χρόνο εισαγωγής στοιχείων στον συγκεκριμένο πίνακα.

Για γρήγορη αναζήτηση χρήστη με βάση όνομα, επώνυμο:

```
create index index_first on User (first_name);
```

```
create index index_last on User (last_name);
```

Για γρήγορη αναζήτηση βιβλίου με βάση τίτλο, διαθέσιμα αντίτυπα:

```
CREATE INDEX index_title ON Book (title);
```

```
CREATE INDEX index_books_number ON Available (books_number);
```

Για γρήγορη αναζήτηση συγγραφέα και θέματος:

```
CREATE INDEX index_author ON Author (name);
```

```
CREATE INDEX index_topic ON Topic (topic);
```

Για γρήγορη αναζήτηση κριτικής με βάση το *username* του χρήστη:

```
CREATE INDEX index_review ON Review (username);
```

Για γρήγορη αναζήτηση δανεισμού με βάση το *username* του χρήστη (είναι καλή ιδέα καθώς ο πίνακας των δανεισμών έχει πολλά δεδομένα και είναι χρήσιμο να μπορεί να γίνει γρήγορα η αναζήτηση δανεισμού για κάποιον χρήστη για να γίνει επιστροφή βιβλίου ακόμα κι αν αυξάνει τον χρόνο εισαγωγής δανεισμού):

```
create index index_start on Borrowing (start_date);
```