

Αναφορά για την Εξαμηνιαία Εργασία στο Μάθημα Βάσεις Δεδομένων

HMMY ΕΜΠ – 6^ο Εξάμηνο

Συντελεστές:

ΝΙΚΟΛΑΟΣ ΣΤΑΜΑΤΟΠΟΥΛΟΣ, AM: 03119020

ΑΘΗΝΑ ΜΑΥΡΟΜΜΑΤΗ, AM: 03119120

LINK για το REPO: https://github.com/ntua-el19020/NTUA_DB_PROJECT_2023

ΔΙΑΓΡΑΜΜΑ ENTITY-RELATIONSHIP:

Το διάγραμμα μπορείτε να το βρείτε στο αρχείο erdiagram.html. Μερικά σχόλια:

- Έχουμε 4 είδη χρηστών, οπότε έχουμε κληρονομική σχέση ανάμεσα στα πεδία Users και GeneralAdmin/SchoolAdmin/Teacher/Student.
- Κάθε σχολική μονάδα έχει στη βιβλιοθήκη της ένα ή περισσότερα βιβλία και κάθε βιβλίο υπάρχει σε μία ή περισσότερες σχολικές μονάδες.
- Οι students μπορούν να κάνουν borrow και reserve βιβλία με όριο 2 για κάθε ενέργεια.
- Οι teachers μπορούν να κάνουν borrow και reserve βιβλία με όριο 1 για κάθε ενέργεια.
- Οι χρήστες κάνουν reservations τα οποία κάνει approve ή deny ο school admin.
- Επίσης, ο school admin εισάγει rentals για τους χρήστες.
- Οι χρήστες μπορούν να δημιουργήσουν reviews για τα βιβλία, τα οποία κάνει approve ή deny ο school admin.

ΣΧΕΣΙΑΚΟ ΔΙΑΓΡΑΜΜΑ:

Το σχεσιακό διάγραμμα μπορείτε να το βρείτε στα αρχεία relationaldiagram.jpg και relationaldiagram.mwb. Μερικά σχόλια:

- Κάθε είδος χρήστη συνδέεται με το table Users και έχει ως foreign key το IdUsers.
- Το table book είναι καθολικό, δηλαδή κάθε βιβλίο που υπάρχει σε όλα τα σχολεία εισάγεται σε αυτό το table και η κάθε σχολική μονάδα ξέρει ποια βιβλία ανήκουν σε αυτήν μέσω του table availability που έχει foreign key το IdSchool. Το table availability περιέχει τα συνολικά αντίτυπα (Copies) που έχει το σχολείο για το συγκεκριμένο βιβλίο και τα αντίτυπα που είναι τώρα διαθέσιμα (AvailableCopies).
- Οι students και οι teachers έχουν τα attributes BooksToBorrow και BooksToReserve τα οποία μεταβάλλονται καταλλήλως όταν αυτοί κάνουν κρατήσεις και δανεισμούς.
- Τα multivariable attributes του ER έχουν γίνει tables (book_writers, book_keywords, book_categories, telephone).

- Τα reviews των χρηστών για ένα βιβλίο καταχωρίζονται στο table review και το Rating στο table book ορίζεται από τον μέσο όρο των RatingLikert για το ίδιο ISBN όπου το review είναι approved.

Ως indexes έχουμε ορίσει τα keys που φαίνονται στο διάγραμμα προκειμένου να επιταχύνουμε τα queries μας.

Η βάση μας περιλαμβάνει επίσης τα triggers:

- update_book_rating: όταν εισάγεται ένα νέο rating υπολογίζεται το συνολικό rating του βιβλίου (αν το rating είναι approved)
- update_book_rating_after_update: όταν αλλάξει ένα rating υπολογίζεται ξανά το συνολικό rating του βιβλίου
- increase_available_copies: αν αλλάξει ο αριθμός των συνολικών αντιτύπων στο availability table, πρέπει να αλλάξει κατά το ίδιο ο αριθμός των διαθέσιμων αντιτύπων
- after_borrowing_insert: όταν εισαχθεί ένα borrowing, BooksToBorrow του user μειώνεται κατά 1
- after_borrowing_update: όταν ένα borrowing αλλάξει (επιστροφή βιβλίου), BooksToBorrow του user αυξάνεται κατά 1
- after_reservation_insert: όταν εισάγεται reservation, BooksToReserve μειώνεται κατά 1 και εάν το reservation είναι Active τότε και το availability μειώνεται κατά 1, αφού τότε το κατοχυρώνει ο χρήστης
- after_reservation_delete: όταν διαγράφεται ένα reservation, BooksToReserve αυξάνεται κατά 1
- reservation_delayed_return: trigger – constraint που πετάει error άμα ο user πάει να κάνει reservation ενώ έχει καθυστερημένη επιστροφή
- reservation_active_borrowing: trigger – constraint που πετάει error άμα ο user πάει να κάνει reservation σε βιβλίο που έχει ήδη αυτή τη στιγμή υπό δανεισμό
- activate_reservations: αν αυξηθεί το availability ενός βιβλίου, τότε κάνει activate την παλαιότερη αίτηση που βρίσκεται στη βάση

Επίσης έχουμε και το event delete_expired_reservations που κάθε μέρα ελέγχει για reservations παλαιότερες από μια εβδομάδα και εάν υπάρχουν τις διαγράφει.