

Βάσεις Δεδομένων

Αναφορά για την εξαμηνιαία εργασία

Δέσποινα Καλλιρρόη Κοσμοπούλου (03120001)

Κωνσταντίνα Μαυρουκάκη-Καραγκούννη (03120063)

Αναστασία Πουλοπούλου (03120074)

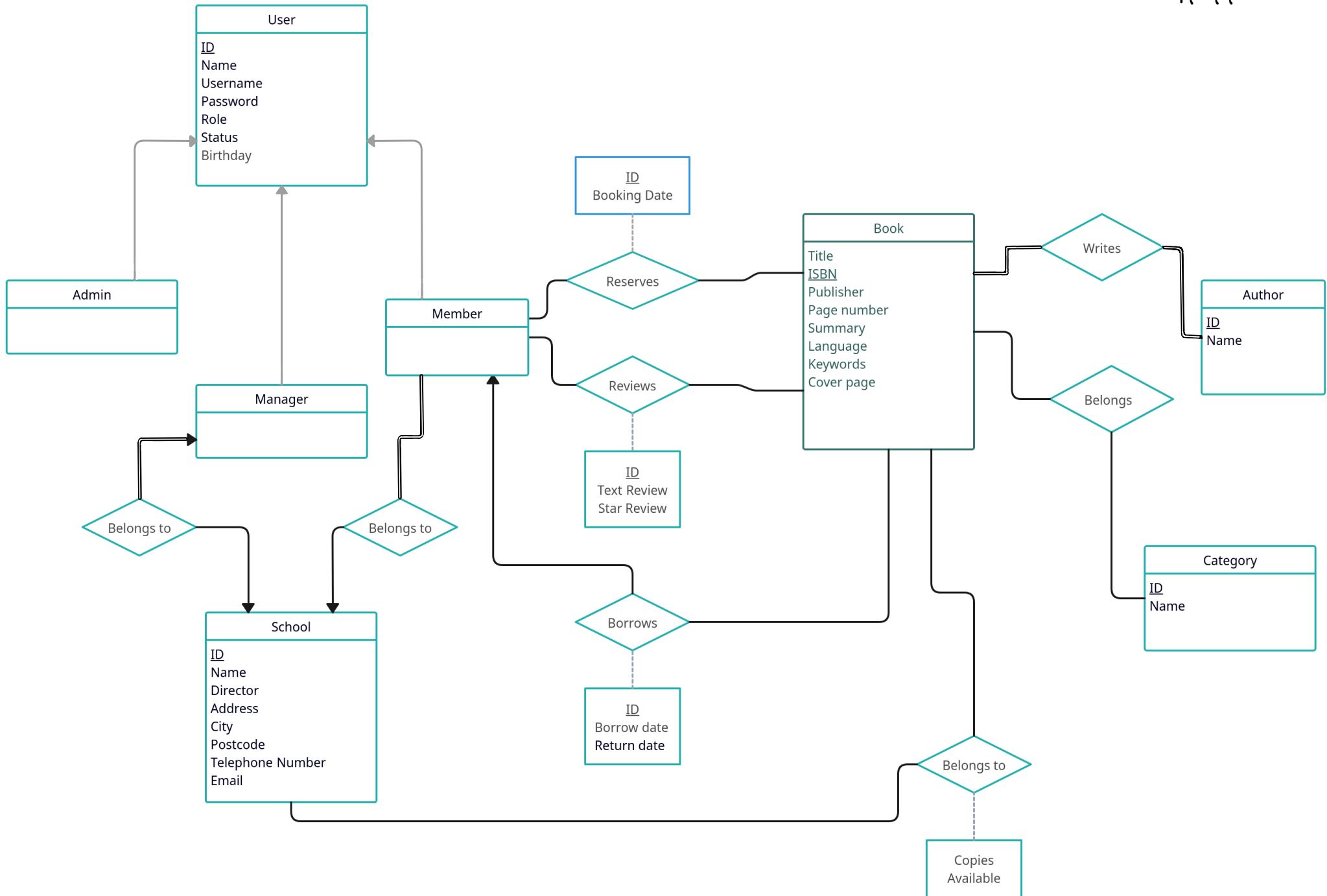
Github repository link: https://github.com/DespoinaKK/school_libraries

Διάγραμμα ER και Σχεσιακό Διάγραμμα

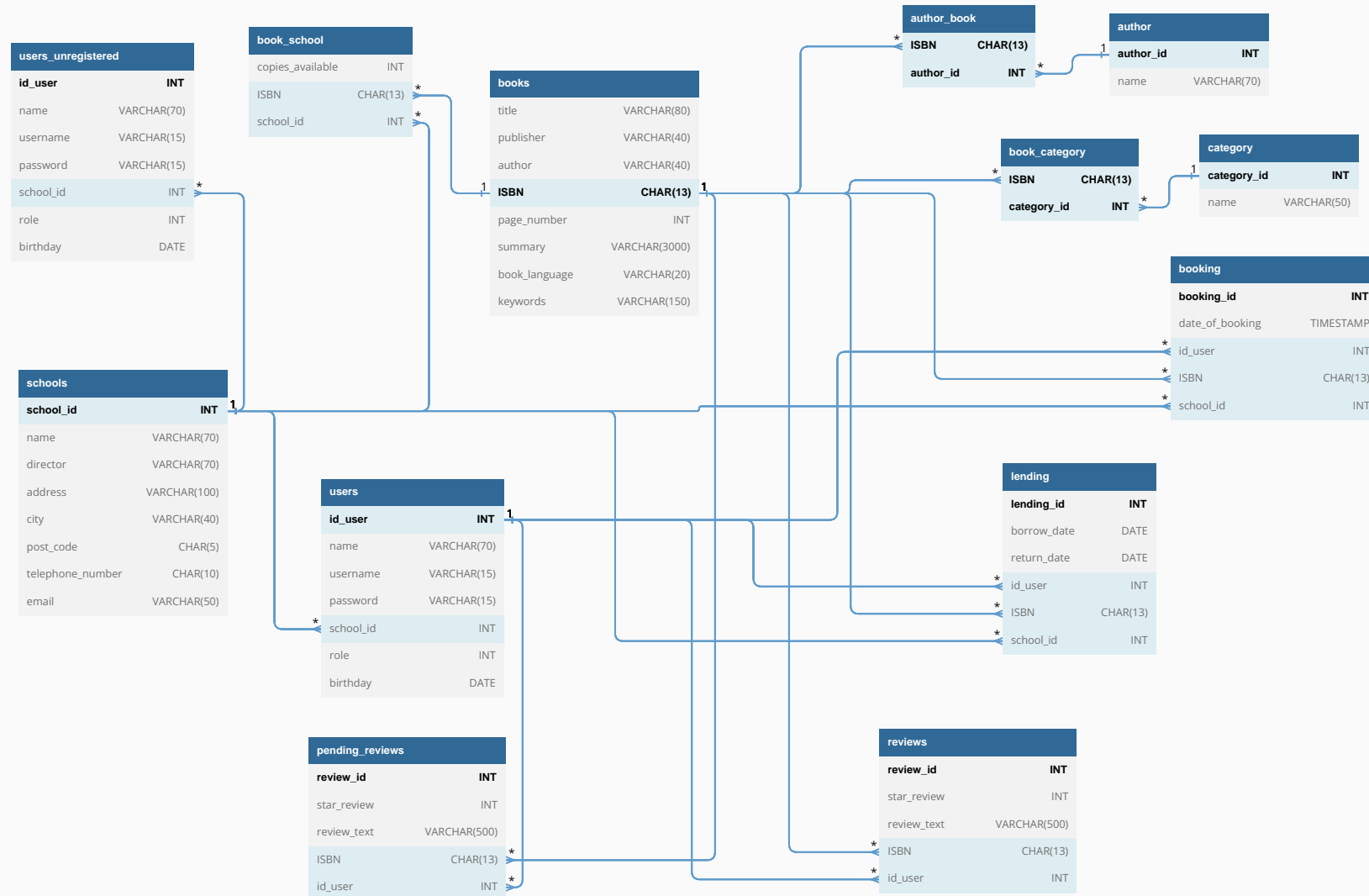
Τα διαγράμματα βρίσκονται στις επόμενες σελίδες.

- Στο ER diagram τα σύνολα οντοτήτων admin, manager, member συντίθενται σε ένα σύνολο οντοτήτων υψηλότερου επιπέδου, την οντότητα user. Στο σχεσιακό διάγραμμα τα παραπάνω υλοποιούνται με την χρήση του πίνακα user και η ιδιότητα διαχωρίζεται με βάση το attribute role (0:member(student), 1:member(teacher), 2:manager, 3:administrator). Με τον τρόπο με τον οποίο υλοποιήσαμε την εφαρμογή υποχρεωτικά με την εγγραφή του κάθε μαθητής, καθηγητής και χειριστής ανήκει σε ένα σχολείο (school_id is not null) ενώ ο διαχειριστής μπορεί να μην ανήκει σε κανένα σχολείο. Κατά την υλοποίηση της εφαρμογής προσθέσαμε τον πίνακα users_unregistered που περιλαμβάνει τους users που η εγγραφή τους δεν έχει γίνει ακόμα αποδεκτή απο τον χειριστή.
- Η οντότητα βιβλίο υλοποιείται με τον πίνακα βιβλίο με primary key το ISBN. Κάθε βιβλίο μπορεί να έχει πολλούς συγγραφείς και κάθε συγγραφέας μπορεί να έχει γράψει πολλά βιβλία, για αυτό υλοποιούμε την σχέση writes με έναν πίνακα author_book. Ομοίως, για τις κατηγορίες, ένα βιβλίο μπορεί να ανήκει σε πολλές κατηγορίες και μια κατηγορία μπορεί να εμπεριέχει πολλά βιβλία, οπότε υλοποιούμε την σχέση belongs με τον πίνακα book_category.
- Ένα βιβλίο μπορεί να ανήκει σε πολλά σχολεία και ένα σχολείο μπορεί να έχει πολλά βιβλία στην βιβλιοθήκη τους. Για αυτό υλοποιούμε την σχέση belongs to με τον πίνακα book_school που περιλαμβάνει και το attribute copies available, το οποίο αναπαριστά τα αντίτυπα που είναι διαθέσιμα την δεδομένη χρονική στιγμή.
- Την συσχέτιση κράτησης την υλοποιήσαμε με ένα πίνακα booking ο οποίος περιλαμβάνει και το timestamp της κράτησης. Ομοίως, για τον δανεισμό, υλοποιήσαμε τον πίνακα lending που περιέχει και τις ημερομηνίες δανεισμού και επιστροφής ως attributes. Η ημερομηνία επιστροφής είναι NULL εως ότου επιστραφεί το βιβλίο.
- Την σχέση reviews την υλοποιήσαμε με τον πίνακα reviews που περιλαμβάνει την αξιολόγηση σε κλίμακα Likert και ελεύθερο κείμενο. Κατά την υλοποίηση της εφαρμογής προσθέσαμε τον πίνακα pending_reviews που περιλαμβάνει τις αξιολογήσεις που δεν έχουν γίνει ακόμα αποδεκτές απο τον χειριστή.
- Για όποιον πίνακα δεν έχει αναφερθεί κάτι διαφορετικό, ως primary key ορίζουμε ένα ID.

Διάγραμμα ER



Σχεσιακό Διάγραμμα



DDL και DML script

Δημιουργήσαμε τους πίνακες χρησιμοποιώντας την εντολή

```
CREATE TABLE table_name (  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
    ....  
);
```

Στην συνέχεια γεμίσαμε την βάση με δεδομένα χρησιμοποιώντας την εντολή

```
INSERT INTO table_name (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...);
```

Το ddl και το dml script υπάρχουν στο [github repo](#)

Triggers

Προσθέσαμε τα παρακάτω triggers:

```
DELIMITER $$  
CREATE TRIGGER `update_copies` AFTER INSERT ON `lending` FOR EACH ROW  
BEGIN  
    UPDATE book_school  
        SET copies_available = copies_available - 1  
        WHERE  
            ISBN = NEW.ISBN;  
END$$  
  
CREATE TRIGGER `return_book` AFTER UPDATE ON `lending` FOR EACH ROW  
BEGIN  
    UPDATE book_school  
        SET copies_available = copies_available + 1  
        WHERE ISBN = OLD.ISBN;  
END$$  
DELIMITER ;
```

Το trigger update_copies αφαιρεί ένα αντίτυπο από τα copies_available στον πίνακα book_school για κάθε δανεισμό που πραγματοποιείται. Το trigger return_book προσθέτει ένα αντίτυπο στα copies_available στον πίνακα book_school για κάθε επιστροφή που πραγματοποιείται

Indexes

Για την καλύτερη λειτουργία της εφαρμογής αποφασίσαμε να προσθέσουμε τα παρακάτω indexes, εκτός των indexes που δημιουργούνται αυτομάτως στα primary και στα foreign keys.

```
CREATE INDEX idx_username ON users (username);
```

```
CREATE INDEX idx_role ON users (role);
```

```
CREATE INDEX idx_booking_date ON booking (date_of_booking);
```

```
CREATE INDEX idx_borrow_date ON lending (borrow_date);
```

```
CREATE INDEX idx_copies ON book_school (copies_available);
```

Τα παραπάνω indexes τα προσθέσαμε σε attributes που χρησιμοποιούνται σε πολλαπλά queries για να αποφεύγεται το επιπλέον κόστος στην αναζήτηση.

Υποθέσεις και επισημάνσεις

Κατά την υλοποίηση της εφαρμογής κάναμε τις παρακάτω υποθέσεις:

- Κάθε user μπορεί να κάνει κράτηση και να δανειστεί βιβλία μόνο από το σχολείο του.
- Η κράτηση μπορεί να πραγματοποιηθεί όταν την δεδομένη στιγμή:
 - Δεν υπάρχουν 2 ενεργές κρατήσεις στην περίπτωση του χρήστη μαθητή και 1 στην περίπτωση του εκπαιδευτικού.
 - Δεν έχει ενεργό δανεισμό ή κράτηση για το ίδιο βιβλίο.
 - Δεν έχει καθύστερήσει την επιστροφή κάποιου βιβλίου με ενεργό δανεισμό.

Οι κρατήσεις μετά από 7 μέρες διαγράφονται αυτόματα.

- Η διαδικασία δανεισμού είναι η ακόλουθη:

Ο χρήστης κάνει κράτηση για ένα βιβλίο και στην συνέχεια ο χειριστής εγκρίνει την κράτηση και καταχωρείται ο δανεισμός. Διαφορετικά ο χειριστής μπορεί να εκτελέσει απευθείας δανεισμό. Αυτομάτως, η εφαρμογή επιτρέπει την καταχώρηση ενός δανεισμού μόνο όταν πληρούνται οι παρακάτω προϋποθέσεις:

 - Ο χρήστης μπορεί, σε διάστημα 7 ημερών, να δανείζεται κατά μέγιστο 2 βιβλία, αν είναι μαθητής, και 1 βιβλίο αν είναι εκπαιδευτικός.
 - Ο δανεισμός δεν μπορεί να καταχωρηθεί αν υπάρχει ενεργός δανεισμός από τον χρήστη που έχει ξεπεράσει το όριο των 7 ημερών.
 - Ο δανεισμός δεν μπορεί να καταχωρηθεί αν ο χρήστης έχει ήδη ενεργό δανεισμό για το ίδιο βιβλίο.

- Υπάρχουν διαθέσιμα αντίτυπα και επαρκούν για την εξυπηρέτηση των κρατήσεων με υψηλότερη χρονική προτεραιότητα.
- Διαγραφή βιβλίων: Κατά την διαδικασία της διαγραφής αφαιρούνται μόνο αντίτυπα από το σχολείο του χειριστή. Ένας χειριστής δεν μπορεί να διαγράψει βιβλία συνολικά από την βάση.
- Διαγραφή-Απενεργοποίηση χρηστών: Ο χειριστής μπορεί να απενεργοποιήσει λογαριασμούς χρηστών. Για λόγους αρχείου το ιστορικό δανεισμού τους και οι αξιολογήσεις τους διατηρούνται στην βάση.
- Επεξεργασία στοιχείων: Ο χρήστης μαθητής μπορεί να αλλάζει μόνο τον κωδικό πρόσβασής του. Ο εκπαιδευτικός μπορεί να αλλάζει όλα τα στοιχεία του εκτός από το όνομα χρήστη. Ο χειριστής και ο διαχειριστής μπορούν να αλλάζουν όλα τους τα στοιχεία.
- Ο κάθε χρήστης μπορεί να αξιολογήσει όλα τα βιβλία του σχολείου του, ακόμα και αν δεν τα έχει δανειστεί.
- Αν ο χειριστής ενός σχολείου επιθυμεί ως εκπαιδευτικός να χρησιμοποιεί τις υπηρεσίες της βιβλιοθήκης θα πρέπει να φτιάξει ξεχωριστό λογαριασμό απλού χρήστη (εκπαιδευτικού).

Οδηγίες Εγκατάστασης της Εφαρμογής

1. Απαραίτητες προϋποθέσεις για την λειτουργία της εφαρμογής είναι να υπάρχουν εγκατεστημένα python3 και mysql.
2. Από το [github repo](#) κατεβάστε τα αρχεία database.sql, app_set.py και τον φάκελο library_app και αποθηκεύστε τα στο ίδιο directory.
3. Συνδεθείτε στην mysql στον υπολογιστή σας και φορτώστε την βάση χρησιμοποιώντας την εντολή:
source {path of file database.sql}
4. Δημιουργήστε ένα virtual environment.
5. Στο ίδιο directory τρέξτε το αρχείο app_set.py (με την εντολή python app_set.py) το οποίο κατεβάζει αυτόματα όλες τις απαραίτητες βιβλιοθήκες και ξεκινά την εφαρμογή. Για την λειτουργία της εφαρμογής πρέπει ο server της βάσης να βρίσκεται σε λειτουργία.
6. Να σημειωθεί ότι σε Windows δεν υπάρχει η δυνατότητα δημιουργίας αντίγραφου ασφαλείας της βάσης και επαναφοράς αυτής μέσω της εφαρμογής.