

## - Εισαγωγή

Η βάση δεδομένων που αποφασίστηκε να γίνει αφορά την βάση δεδομένων μιας βιβλιοθήκης. Επιλέχθηκε αυτή η βάση συνολικά από την ομάδα, γιατί πιστεύαμε ότι μπορούμε να βγάλουμε ένα αρκετά δημιουργικό αποτέλεσμα, όχι μόνο στην θεματική των βάσεων, αλλά και στην θεματική των **procedures** και των **triggers**.

Προφανώς, όλες οι διαθέσιμες θεματικές ήταν ικανοποιητικές, αλλά η βάση δεδομένων μιας βιβλιοθήκης μας τράβηξε, εν τέλη, το ενδιαφέρον.

## - Περιγραφή των tables, των triggers και των procedures

### Η δομή των **πινάκων**:

Υπάρχουν 5 πίνακες στην βάση. Ο **reservations**, **books**, **customers**, **staff** και **subscriptions** καθώς και ένας πίνακας **res\_logs** για την δημιουργία **log file**

-- Το **table reservations** περιέχει

- το πεδίο **RID** τύπου **int**. Η συγκεκριμένη μεταβλητή είναι **Κύριο κλειδί** στον πίνακα και **Not Null**.
- το πεδίο **CID** τύπου **int** και **Not Null**.
- το πεδίο **SID** τύπου **int** και **Not Null**.
- το πεδίο **BID** τύπου **int** και **Not Null**.
- το πεδίο **Exp\_date** τύπου **timestamp** και **Not Null**.

Οι μεταβλητές **CID**, **SID**, **BID** αποτελούν **Ξένα κλειδιά** του πίνακα από τους πίνακες **customers**, **staff** και **books** αντίστοιχα που θα δούμε παρακάτω

-- Το **table books** περιέχει

- το πεδίο **BID** τύπου **int**. Η συγκεκριμένη μεταβλητή είναι **Κύριο κλειδί** και **Not Null**.
- το πεδίο **Name** τύπου **varchar** και **Not Null**.
- το πεδίο **Genre** τύπου **varchar** και **Not Null**.
- το πεδίο **Publisher** τύπου **varchar** και **Not Null**.
- το πεδίο **Writer** τύπου **varchar** και **Not Null**.
- το πεδίο **Quantity** τύπου **int** και **Not Null**. Η συγκεκριμένη μεταβλητή έχει περιορισμό **CHECK (Quantity > -1)** ώστε το πλήθος των βιβλίων να είναι πάντα φυσικός αριθμός.

-- Το **table customers** περιέχει

- το πεδίο **CID** τύπου **int**. Η συγκεκριμένη μεταβλητή είναι **Κύριο κλειδί** και **Not Null**.
- το πεδίο **Name** τύπου **varchar** και **Not Null**.
- το πεδίο **Contacts** τύπου **varchar** και **Not Null**.
- το πεδίο **DOB** τύπου **year** και **Not Null**.
- το πεδίο **Sub\_ID** τύπου **int**. Η συγκεκριμένη μεταβλητή είναι **Ξένο κλειδί** από τον πίνακα **subscriptions** και **Not Null**.

-- Το **table staff** περιέχει

--- το πεδίο **SID** τύπου **INT**. **int**. Η συγκεκριμένη μεταβλητή είναι **Κύριο κλειδί** και **Not Null**.  
--- το πεδίο **Name** τύπου **varchar** και **Not Null**.  
--- το πεδίο **Job** τύπου **varchar** και **Not Null**.  
--- το πεδίο **Contacts** τύπου **varchar** και **Not Null**.

-- Το **table subscriptions** περιέχει

--- το πεδίο **Sub\_ID** τύπου **int**. Η συγκεκριμένη μεταβλητή είναι **Κύριο κλειδί** και **Not Null**.  
--- το πεδίο **Duration** τύπου **int** και **Not Null**.  
--- το πεδίο **Price** τύπου **double** και **Not Null**.  
--- το πεδίο **Types** τύπου **varchar** και **Not Null**.

-- Το **table res\_logs**, το οποίο περιέχει

--- το πεδίο **Command** τύπου **varchar** και **Not Null**.  
--- το πεδίο **RID** τύπου **int** και **Not Null**.  
--- το πεδίο **CID** τύπου **int** και **Not Null**.  
--- το πεδίο **BID** τύπου **int** και **Not Null**.  
--- το πεδίο **SID** τύπου **int** και **Not Null**.  
--- το πεδίο **Exp\_date** τύπου **timestamp** και **Not Null**.  
--- το πεδίο **StartDate** τύπου **datetime** και **Not Null**.  
--- το πεδίο **EndDate** τύπου **datetime** και **Not Null**.

Τα πεδία **Command**, **RID**, **StartDate** και **EndDate** αποτελούν το **Κύριο κλειδί** του πίνακα.

Τα **triggers** που χρησιμοποιήθηκαν είναι:

-- **Create** του **trigger HistoryTableInsert**, το οποίο κάνει **insert res\_logs** τις **Τιμές** ('Inserted', New.RID, New.CID, New.BID, New.SID, New.Exp\_date, N, NULL), όταν γίνεται κάποιο **insert** στον πίνακα **reservation**.

--- Η τιμή **N** τύπου **datetime** γίνεται **declare** μέσα στο **trigger** και η τιμή ορίζεται μέσω της μεθόδου **now()**.

-- **Create** του **trigger HistoryTableDelete**, το οποίο κάνει **update** στο **res\_logs**

--- Η τιμή **N** τύπου **datetime** γίνεται **declare** μέσα στο **trigger** και η τιμή ορίζεται μέσω της μεθόδου **now()**.  
--- Όταν γίνεται **delete** στον πίνακα **reservation**, τότε γίνεται **update** στον **res\_logs**, κάνοντας **set** το **EndDate** ως **N** και το **Command** ως 'Deleted', όπου το **RID = Old.RID** και **BID = Old.BID** και **SID = Old.SID** και **EndDate IS NULL**.

-- **Create** του **trigger HistoryTableUpdate**, το οποίο κάνει **update** στην παλιά καταγραφή και νέο **insert** στο **res\_logs** τις **Τιμές** ('Updated', New.RID, New.CID, New.BID, New.SID, New.Exp\_date, N, NULL)

--- Η τιμή **N** τύπου **datetime** γίνεται **declare** μέσα στο **trigger** και η τιμή ορίζεται μέσω της μεθόδου **now()**.  
--- Όταν γίνεται **update** στον πίνακα **reservation**, τότε γίνεται **update** στον **res\_logs**, κάνοντας **set** το **EndDate** ως **N** και το **Command** ως 'Updated', όπου το **RID = Old.RID** και **BID = Old.BID** και **SID = Old.SID** και **EndDate IS NULL**

Τα **procedures** που χρησιμοποιήθηκαν είναι:

#### Get:

-- **get\_res\_logs()** όπου επιστρέφει το περιεχόμενο του **table res\_logs**.  
  
-- **get\_Staff()** όπου επιστρέφει το περιεχόμενο του **table staff**.  
  
-- **get\_Customers()** όπου επιστρέφει διαλέγει το περιεχόμενο του **table customers**.  
  
-- **get\_Books()** όπου επιστρέφει το περιεχόμενο του **table books**.  
  
-- **get\_Reservations()** όπου επιστρέφει το περιεχόμενο του **table reservations**.  
  
-- **get\_Subs()** όπου επιστρέφει το περιεχόμενο του **table subscriptions**.  
  
-- **get\_StaffWithNameLike**(in onoma varchar(45)) όπου επιστρέφει τις σειρές του **table staff** με **name** που περιέχει την μεταβλητή **onoma**.  
  
-- **get\_CustomersWithNameLike**(in onoma varchar(45)) όπου επιστρέφει τις σειρές του **table customers** με **name** που περιέχει την μεταβλητή **onoma**.  
  
-- **get\_BooksWithNameLike**(in onoma varchar(45)) όπου επιστρέφει τις σειρές του **table books** με **name** που περιέχει την μεταβλητή **onoma**.

#### Add:

-- **addSub**(in type varchar(45), in Duration int(11), in Price double, in Sub\_ID int(11))  
Το συγκεκριμένο προσθέτει στο **table subscriptions** τα στοιχεία που έχουν δοθεί ως παράμετροι.  
  
-- **addCustomer**(IN name VARCHAR(45), IN cid INT(11), IN contacts VARCHAR(45), IN dob YEAR(4), IN sub\_id INT(11))  
Το συγκεκριμένο προσθέτει στο **table customers** τα στοιχεία που έχουν δοθεί ως παράμετροι.  
  
-- **addStaff**(IN name VARCHAR(45), IN sid INT(11), IN job VARCHAR(45), IN contacts VARCHAR(45))  
Το συγκεκριμένο προσθέτει στο **table staff** τα στοιχεία που έχουν δοθεί ως παράμετροι.

-- **addReservation**(IN rid int(11),IN cid INT(11), IN bid INT(11), IN sid INT(11), IN exp\_date  
TIMESTAMP)

Το συγκεκριμένο προσθέτει στο **table reservations** τα στοιχεία που έχουν δοθεί ως παράμετροι. Αυτό που κάνει, επίσης, είναι να μειώνει κατά ένα το **quantity** του βιβλίου το οποίο έχει επιλεχθεί για κράτηση.

-- **addBook**(IN name VARCHAR(45), IN bid INT(11), IN genre VARCHAR(45), IN publisher  
VARCHAR(45), IN writer VARCHAR(45), IN quantity INT(11))

Το συγκεκριμένο προσθέτει στο **table books** τα στοιχεία που έχουν δοθεί ως παράμετροι.

### Delete:

-- **deleteCustomer**(IN ciid INT(11))

Με βάση το **ciid**, γίνεται διαγραφή από το **table customers**, η σειρά όπου το **ciid** αντιστοιχεί με το **CID**.

-- **deleteStaff**(IN siid INT(11))

Με βάση το **siid**, γίνεται διαγραφή από το **table staff**, η σειρά όπου το **ciid** αντιστοιχεί με το **SID**.

-- **deleteReservation**(IN riid INT(11))

Με βάση το **riid**, γίνεται διαγραφή από το **table reservation**, η σειρά όπου το **ciid** αντιστοιχεί με το **RID**. Αυτό που κάνει, επίσης, είναι να προσθέτει ένα στο **quantity** του βιβλίου το οποίο έχει επιστραφεί.

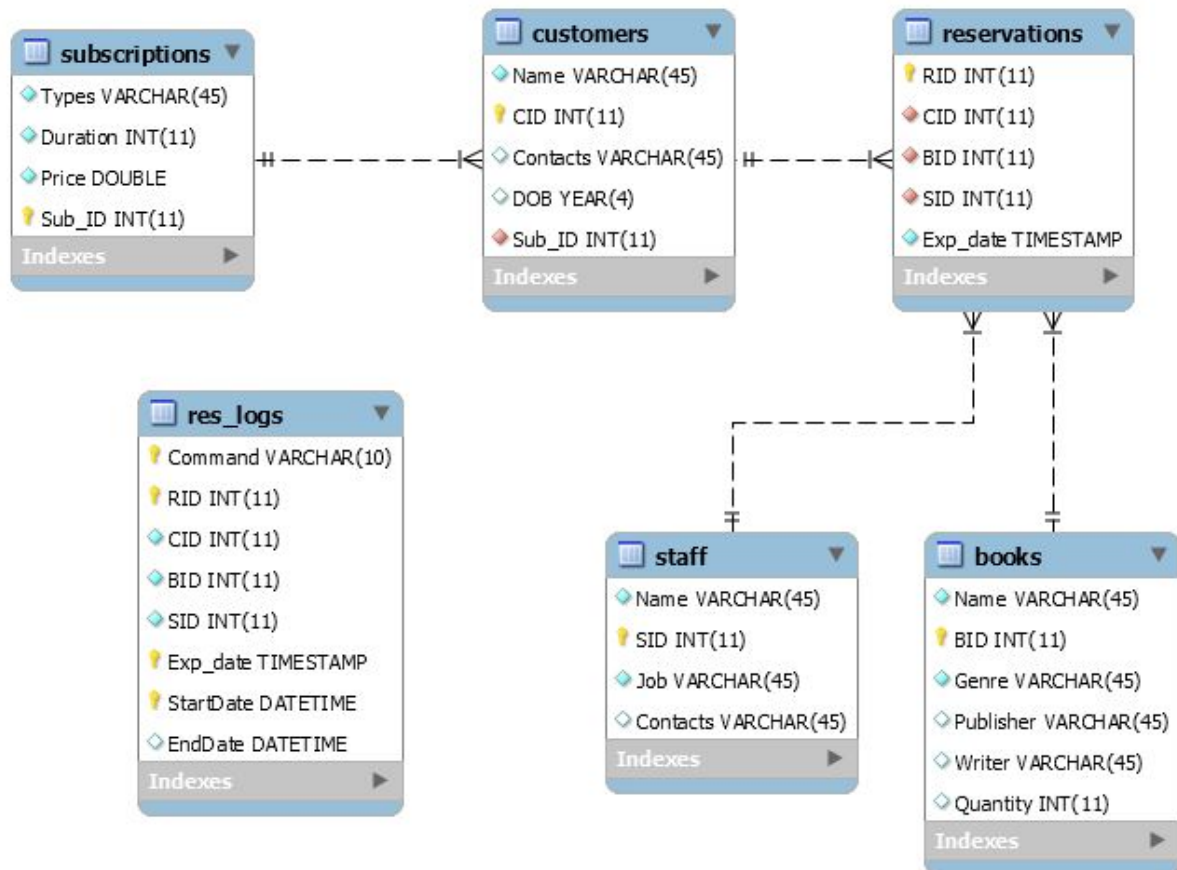
-- **deleteBook**(IN biid INT(11))

Με βάση το **biid**, γίνεται διαγραφή από το **table books**, η σειρά όπου το **biid** αντιστοιχεί με το **BID**.

-- **deleteSub**(IN sub\_iid int)

Με βάση το **sub\_iid**, γίνεται διαγραφή από το **table subscriptions**, η σειρά όπου το **sub\_iid** αντιστοιχεί με το **Sub\_ID**.

Το ER της βάσης είναι:



## - Εφαρμογή (Java Swing, γενικές διαδικασίες)

Το πρόγραμμα χωρίζεται σε 7 κλάσεις, από τις οποίες οι 6 καλούνται από ένα αντίστοιχο **button**. Αυτές οι έξι κλάσεις, πέραν της κλάσης **MySQLConnection**, κάνουν extend το **javaw.swing.JFrame**. Τα **buttons** αυτά παρουσιάζονται όλα στην κλάση **MainMenu()**, από την οποία και καλούνται οι υπόλοιπες κλάσεις.

### class MainMenu

Το κύριο μενού (**class MainMenu**) είναι η κλάση που περιέχει την **main()** και θα είναι το πρώτο panel που θα δει ο χρήστης όταν τρέξει το πρόγραμμα. Στην **main()**, υπάρχουν **6 buttons**, **5** τα οποία αφορούν τα υπάρχοντα **tables**, και ένα το οποίο αφορά το **log file**.

Τα **ButtonActionPerformed** που αφορούν τα **tables** είναι:

- **ListalpallilonButtonActionPerformed**(java.awt.event.ActionEvent evt), το οποίο δημιουργεί το αντικείμενο **ipalliloi** τύπου **Listalpallilon** και το κάνει **visible**.
- **ListaPelatonButtonActionPerformed**(java.awt.event.ActionEvent evt), το οποίο δημιουργεί το αντικείμενο **pelates** τύπου **ListaPelaton** και το κάνει **visible**.
- **KratiseisButtonActionPerformed**(java.awt.event.ActionEvent evt), το οποίο δημιουργεί το αντικείμενο **kratiseis** τύπου **Kratiseis** και το κάνει **visible**.
- **ApothemataButtonActionPerformed**(java.awt.event.ActionEvent evt), το οποίο δημιουργεί το αντικείμενο **apothemata** τύπου **ListaBiblion** και το κάνει **visible**.
- **SubTypeButtonActionPerformed**(java.awt.event.ActionEvent evt), το οποίο δημιουργεί το αντικείμενο **eggrafes** τύπου **ListaEggrafon** και το κάνει **visible**.

Όσον αφορά το **log file**, έχουμε την μέθοδο **refreshLogsButtonActionPerformed**(java.awt.event.ActionEvent evt), η οποία καλεί την **backlogger()**. Μέσα από την **backlogger()**, γίνεται εγγραφή στο αρχείο **logger.txt** οι αλλαγές που έχουν γίνει στον πίνακα **res\_logs**, μέσω της μεθόδου **getLogs()**.

Από το **menu**, πατώντας το κουμπί ανάλογα με τις ανάγκες που έχει ο χρήστης, ανοίγει το εκάστοτε παράθυρο

### **class Kratiseis, ListaBiblion, ListaEggrafon, Listalpallilon, ListaPelaton**

Όλες οι κλάσεις ξεκινούν με την δημιουργία αντικειμένου **mysqlConnect** τύπου **MySQLConnect** μέσω της **class MySQLConnect()**.

Σε όλες τις κλάσεις, δημιουργούνται:

-- Μία φόρμα, όπου περιέχει κάποια **text boxes** για την εκχώρηση των κατάλληλων δεδομένων από τον χρήστη.

--- Στην κλάση **ListaPelaton** στην θέση της επιλογής των **subscriptions** έχει δημιουργηθεί ένα **combo box** (**dynamicComboBox()**) για να υπάρχει συγκεκριμένη επιλογή για τον τύπο εγγραφής των πελατών το οποίο γεμίζει δυναμικά από τα περιεχόμενα του πίνακα **subscriptions**.

-- Τρία buttons, τα οποία καλούν τα αντίστοιχα **procedures**, ανάλογα την κλάση που βρίσκεται ο χρήστης:

--- **Insert** (**InsertButtonActionPerformed**(java.awt.event.ActionEvent evt))

--- **Delete** (**DeleteButtonActionPerformed**(java.awt.event.ActionEvent evt))

--- **RefreshTable**

(**refreshButtonActionPerformed**(java.awt.event.ActionEvent evt))

-- Έναν πίνακα με πεδία αντίστοιχα του κάθε πίνακα που προβάλλει η κλάση.

-- Για τις κλάσεις **ListaPelaton**, **ListaBiblion** και **Listalpallilon** έχει τοποθετηθεί ένα **search bar** (**searchButtonActionPerformed**(java.awt.event.ActionEvent evt)). Σε αυτό, ο χρήστης μπορεί να αναζητήσει δυναμικά το **name** το οποίο θέλει να βρει από τον εκάστοτε πίνακα που προβάλλει η αντίστοιχη κλάση.

### **Οι κοινές μέθοδοι σε κάθε κλάση είναι:**

-- Με την μέθοδο **InsertButtonActionPerformed**(java.awt.event.ActionEvent evt), δίνοντας ο χρήστης τις τιμές των **text boxes**, οι τιμές εισάγονται στην βάση δημιουργώντας ένα **statement**, το οποίο εκτελεί το αντίστοιχο **procedure** τύπου **add**, περνώντας παραμετρικά τις παρακάτω μεταβλητές.

Τα στοιχεία που δίνονται στην εκάστοτε φόρμα είναι:

--- Για την κλάση **Listalpallilon** έχουμε τα **text boxes** που αντιστοιχούν στις **String** μεταβλητές

---- Η μεταβλητή **String name** θα πάρει την τιμή από το **nameField.getText()**

---- Η μεταβλητή **String sid** θα πάρει την τιμή από το **sidField.getText()**

---- Η μεταβλητή **String job** θα πάρει την τιμή από το **jobField.getText()**

---- Η μεταβλητή **String contact** θα πάρει την τιμή από το **contactField.getText()**

--- Για την κλάση **ListaPelaton** έχουμε τα **text boxes** που αντιστοιχούν στις **String** μεταβλητές

---- Η μεταβλητή **String name** θα πάρει την τιμή από το **nameField.getText()**  
---- Η μεταβλητή **String cid** θα πάρει την τιμή από το **cidField.getText()**  
---- Η μεταβλητή **String contact** θα πάρει την τιμή από το **contactField.getText()**  
---- Η μεταβλητή **String dob** θα πάρει την τιμή από το **dobField.getText()**  
---- Η μεταβλητή **int subCombo** θα πάρει την τιμή από το **combo box** που προαναφέρθηκε.

--- Για την κλάση **Kratiseis** έχουμε τα **text boxes** που αντιστοιχούν στις **String** μεταβλητές

---- Η μεταβλητή **String rid** θα πάρει την τιμή από το **ridField.getText()**  
---- Η μεταβλητή **String cid** θα πάρει την τιμή από το **cidField.getText()**  
---- Η μεταβλητή **String bid** θα πάρει την τιμή από το **bidField.getText()**  
---- Η μεταβλητή **String sid** θα πάρει την τιμή από το **sidField.getText()**  
---- Η μεταβλητή **String expdate** θα πάρει την τιμή από το **expdateField.getText()**

--- Για την κλάση **ListaBiblion** έχουμε τα **text boxes** που αντιστοιχούν στις **String** μεταβλητές

---- Η μεταβλητή **String name** θα πάρει την τιμή από το **nameField.getText()**  
---- Η μεταβλητή **String bid** θα πάρει την τιμή από το **bidField.getText()**  
---- Η μεταβλητή **String genre** θα πάρει την τιμή από το **genreField.getText()**  
---- Η μεταβλητή **String publisher** θα πάρει την τιμή από το **publisherField.getText()**  
---- Η μεταβλητή **String writer** θα πάρει την τιμή από το **writerField.getText()**  
---- Η μεταβλητή **String quantity** θα πάρει την τιμή από το **quantityField.getText()**  
---- Η μεταβλητή **String Sub\_ID** θα πάρει την τιμή από το **Sub\_IDField.getText()**

--- Για την κλάση **ListaEggrafon** έχουμε τα **text boxes** που αντιστοιχούν στις **String** μεταβλητές

---- Η μεταβλητή **String Type** θα πάρει την τιμή από το **TypeField.getText()**  
---- Η μεταβλητή **String Duration** θα πάρει την τιμή από το **DurationField.getText()**  
---- Η μεταβλητή **String Price** θα πάρει την τιμή από το **PriceField.getText()**  
---- Η μεταβλητή **String Sub\_ID** θα πάρει την τιμή από το **Sub\_IDField.getText()**

--- Αν τα στοιχεία είναι σωστά, εισαχθούν στην βάση και δεν δημιουργηθεί **exception**, εμφανίζεται στον χρήστη ένα μήνυμα επιβεβαίωσης σε ένα **dialogue box** και ο κώδικας αρχικοποιεί την κατάσταση των **text boxes**.

--- Αν δημιουργηθεί **exception**, τότε εμφανίζεται το **exception message** στον χρήστη και τα στοιχεία παραμένουν ίδια στα **text boxes**.

-- Με την μέθοδο **DeleteButtonActionPerformed(java.awt.event.ActionEvent evt)**, επιλέγεται το **id** το οποίο έχει δοθεί παραμετρικά στο αντίστοιχο **text box**. Αφού γίνει αυτό, αφαιρείται από την βάση το **row**, δημιουργώντας ένα **statement**, το οποίο εκτελεί το

αντίστοιχο **procedure** τύπου **delete**, περνώντας παραμετρικά μια από τις παρακάτω μεταβλητές. Η αφαίρεση ενός **row** από την βάση γίνεται μόνο από το αντίστοιχο **id** της.

--- Αν τα στοιχεία είναι σωστά, αντιστοιχιστεί το **id** που δόθηκε δυναμικά με κάποιο **id** που υπάρχει στην βάση και δεν δημιουργηθεί **exception**, εμφανίζεται στον χρήστη ένα μήνυμα επιβεβαίωσης σε ένα **dialogue box** και ο κώδικας αρχικοποιεί την κατάσταση των **text boxes**.

--- Αν δημιουργηθεί **exception**, τότε εμφανίζεται το **exception message** στον χρήστη και τα στοιχεία παραμένουν ίδια στα **text boxes**.

-- Με την μέθοδο **refreshButtonActionPerformed(java.awt.event.ActionEvent evt)**, γίνεται ανανέωση του πίνακα και εμφανίζονται οι νεότερες προσθήκες ή αφαιρέσεις του εκάστοτε πίνακα. Όλο αυτό συμβαίνει δημιουργώντας ένα **statement** και καλώντας το αντίστοιχο **procedure** τύπου **Get** από την βάση. Σε περίπτωση που δημιουργηθεί κάποιο **exception**, τότε εμφανίζεται στον χρήστη το αντίστοιχο **exception message**.

### class MySQLConnection

Η κλάση αυτή χρησιμοποιείται για την σύνδεση στην βάση. Δηλώνονται ως τύπου **String** ο **DATABASE\_DRIVER**, το **DATABASE\_URL**, το **USERNAME**, το **PASSWORD** και το **MAX\_POOL** της βάσης, τα οποία περνώνται παραμετρικά ως **Properties** του **Connection**. Υπάρχουν επίσης οι εκάστοτε κλάσεις **connect()** και **disconnect()**, οι οποίες είναι υπεύθυνες για την σύνδεση ή την αποσύνδεση από την βάση.

### Βιβλιοθήκες

Στο πρόγραμμα χρειάστηκε ο **Driver Manager mysql-connector-java-8.0.22** που είναι υπεύθυνος για την συνδεσιμότητα με την βάση καθώς είναι στημένη σε **MySQL**.

### - Συμπεράσματα

Υπήρχε συνεργασία στην ομάδα σε έναν αρκετά επαρκή βαθμό. Η συνεννόηση έγινε μέσω **voice chat** τακτικά, αλλά ο πιο συνηθισμένος τρόπος ήταν μέσω **chat**. Η χρήση του **voice chat** έγινε ώστε να δίνουμε ένα **self-report** στους συναδέλφους μας, καθώς και να συζητηθεί ποιο μέρος της εργασίας θα πρέπει να ολοκληρωθεί μέχρι την επόμενη συνάντηση.

Οι δουλειές μοιράστηκαν αντάξια και ο καθένας αν έπαιρνε παραπάνω δουλειά, θα ήταν καθαρά και μόνο αν το αντικείμενο ήταν εντελώς άγνωστο προς κάποιον άλλον. Ουσιαστικά γινόταν κυρίως για διευκόλυνση, αλλά πάντα το τελικό αποτέλεσμα υποστηριζόταν και από τα τρία μέλη της ομάδας.

Αν δοθεί η ευκαιρία, η ομάδα θα ήθελε να ξαναβρεθεί σε αυτήν την μορφή, μιας και δεν είχαμε παράπονο ή κάποια διαφωνία που θα κόστιζε το ομαδικό πνεύμα. Υπήρχε, εν τέλει, μία άψογη συνεργασία.

Τα μέλη της ομάδας,

Κόνσουλας Ηρακλής  
Τράντος Χρυσόστομος Αλέξανδρος  
Ψαρράς Αθανάσιος



