

## ΕΡΓΑΣΙΑ ΣΧΕΔΙΑΣΜΟΥ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

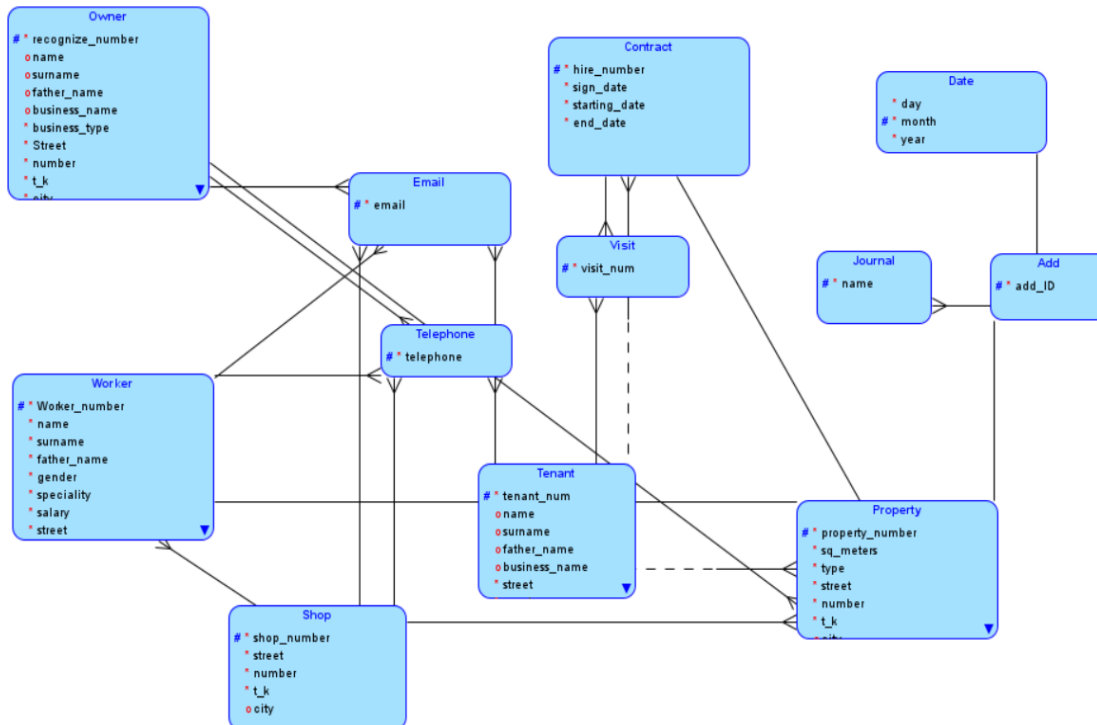
Όνομα : Χατζόπουλος Ευθύμιος

Αριθμός Μητρώου : E21187

email : [efhtimishatzo@gmail.com](mailto:efhtimishatzo@gmail.com)

(Ο Κώδικας αποτελείται από τις δύο κλάσεις Main.java, MySqlConnection.java που βρίσκονται στον φάκελο database\_project → src .)

## • Λογικό Μοντέλο:

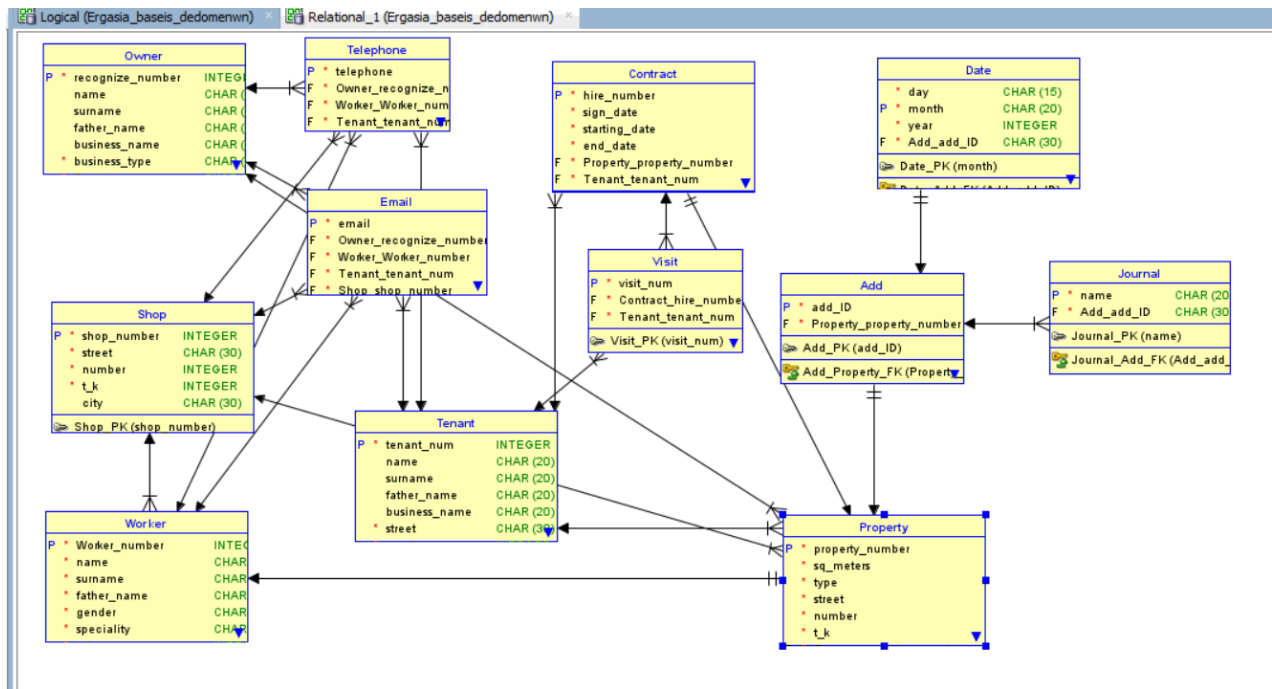


Στο παραπάνω λογικό μοντέλο έχουν γίνει οι εξής παραδοχές:

1. Έχουμε φτιάξει οντότητες "Email", "Telephone" με τις οποίες οι οντότητες "Owner", "Worker", "Shop" και "Tenant" έχουν σχέση 1:N αφού στην εκφώνηση τονίζεται ότι η κάθε οντότητα μπορεί να έχει από 1 έως πολλά τηλέφωνα.
2. Οι οντότητες "Owner", "Property" έχουν σχέση 1:N αφού κάθε ιδιοκτήτης μπορεί να έχει 1 ή πολλά ακίνητα.
3. Οι οντότητες "Add", "Property" έχουν σχέση 1:1 αφού σε κάθε αγγελία δημοσιεύεται ένα ακίνητο.
4. Οι οντότητες "Add", "Date" έχουν σχέση 1:1 αφού κάθε αγγελία έχει μία ημερομηνία.
5. Οι οντότητες "Add", "Journal" έχουν σχέση 1:N καθώς μία αγγελία μπορεί να δημοσιευθεί σε πολλές εφημερίδες.
6. Οι οντότητες "Contract", "Property" έχουν σχέση 1:1 αφού με κάθε συμβόλαιο νοικιάζεται ένα ακίνητο.
7. Οι οντότητες "Contract", "Tenant" έχουν σχέση N:1 που στην πλευρά του ενοικιαστή μπορεί να είναι και 0 καθώς μπορεί κάποιος ενοικιαστής να μην έχει νοικιάσει ακόμα ακίνητο.
8. Οι οντότητες "Contract", "Tenant" έχουν σχέση 1:N με την "Visit" αφού κάθε ενοικιαστής μπορεί να επισκεφτεί πολλές φορές το ακίνητο πριν υπογράψει το συμβόλαιο.
9. Οι οντότητες "Shop", "Property" έχουν σχέση 1:N επειδή κάθε ακίνητο διαχειρίζεται από ένα υποκατάστημα, αλλά ένα υποκατάστημα μπορεί να διαχειρίζεται πολλά ακίνητα.

10. Οι οντότητες “Shop”, “Worker” έχουν σχέση 1:N αφού κάθε υπάλληλος ανήκει σε ένα υποκατάστημα αλλά κάθε υποκατάστημα μπορεί να διαθέτει πολλούς υπαλλήλους .
11. Οι οντότητες “Property”, “Worker” έχουν σχέση 1:1 διότι κάθε υπάλληλος αναλαμβάνει ένα ακίνητο.

- **Σχεσιακό Μοντέλο:**



- **Σκεπτικό Java Εφαρμογής & Οδηγίες Χρήσης:**

Με την εκκίνηση της εφαρμογής εμφανίζεται το παρακάτω menu:

```

1
***** MENU *****
1. Insert or Update the Data Base
2. Show the list with every property available per shop
3. Show every property with an active contract from all the shops
4. Show the list of properties available for renting for a particular shop
5. Make a contract for property renting
6. Delete every unavailable property for renting
7. Show every unrented property from every shop which is published in the newspaper
8. Show every possible tenant who visited a property more than once
9. Update email, telephone number or address
10. Exit Menu
1

```

Η κάθε επιλογή αποτελεί και ένα από τα ερωτήματα της εργασίας. Επομένως, ο χρήστης εισάγει από 1 έως 10 για να επιλέξει.

1. Με την επιλογή του 1 εμφανίζεται ένα νέο Menu και εισάγονται στην βάση αυτόματα ,εάν δεν υπάρχουν ήδη, ένας ενοικιαστής , ένας ιδιοκτήτης, ένας υπάλληλος και ένα μαγαζί με αριθμούς 0. (Βοηθάνε στην εισαγωγή property, email, telephone):

```
1
***** OPTIONS *****
1. Insert Tenant
2. Insert Property
3. Insert Rental
4. Update Tenant
5. Update Propety
6. Update Rental
<
```

Σύμφωνα με το οποίο ο χρήστης έχει την επιλογή εισάγοντας 1 – 6 να εισάγει ή να αναβαθμίσει ενοικιαστή , ακίνητο ή συμβόλαιο.

Οποιαδήποτε επιλογή κι αν κάνει ο χρήστης από τις έξι το πρόγραμμα ζητάει εισαγωγή αριθμού ενοικιαστή , ακινήτου ή συμβόλαιο. Ελέγχεται εάν υπάρχουν ήδη με τη χρήση της `conn.select(entity, variable, value)` και αν ναι εμφανίζεται σχετικό μήνυμα.

Στην εισαγωγή στην περίπτωση του ενοικιαστή ζητώνται τα στοιχεία του και έπειτα εισάγονται στη βάση.

Στην περίπτωση που θέλουμε να εισάγουμε ακίνητο το πρόγραμμα ελέγχει επιπλέον εάν υπάρχει ο αριθμός ιδιοκτήτη , ο αριθμός εργάτη και ο αριθμός υποκαταστήματος ώστε να μην γίνει παραβίαση ξένων κλειδιών. Εάν δεν υπάρχουν ζητάει από τον χρήστη τα στοιχεία τους και τα εισάγει τη βάση. Σε κάθε εισαγωγή νέου ακινήτου θέτουμε το `property_tenant_fk = 0` για να δηλώσουμε ότι η επιχείρηση δεν έχει νοικιαστεί ακόμα.

Παράδειγμα για “Owner”

```

Connect.java Main.java X
System.out.println("Please enter owner's number:\n");
int o_num = sc.nextInt(); // enter owner number

// searches if owner's number already exists
rs1 = conn.select("owner", "recognize_number", o_num);

if (rs1 == null) { // if owner doesn't already exist
    System.out.println("Owner not found! please insert the following attributes:\n");

    System.out.println("Please enter owner's name or null if it is a business:\n");
    String o_name = sc.next(); // enter owner name
    System.out.println("Please enter owner's surname or null if it is a business:\n");
    String o_surname = sc.next(); // enter owner surname
    System.out.println("Please enter owner's fathername or null if it is a business:\n");
    String o_fname = sc.next(); // enter owner father name
    System.out.println("Please enter owner's business name or null if it is a person:\n");
    String o_bname = sc.next(); // enter business name
    System.out.println("Please enter owner's business type or null if it is a person:\n");
    String o_btype = sc.next(); // enter business type
    System.out.println("Please enter owner's address street (instead of space use _):\n");
    String o_street = sc.next(); // enter street from address
    System.out.println("Please enter owner's address number:\n");
    int o_addr_num = sc.nextInt(); // enter owner's address number
    System.out.println("Please enter owner's address T.K:\n");
    int o_t_k = sc.nextInt(); // enter owner's address T.K: taxydomikos kodikas
    System.out.println("Please enter owner's city of residence:\n");
    String o_city = sc.next(); // enter city of residence
    System.out.println("Please enter owner's email:\n");
    String o_email = sc.next(); // enter email
    System.out.println("Please enter owner's telephone number:\n");
    int o_tel = sc.nextInt(); // enter owner's telephone number

    conn.insertOwner(o_num, o_name, o_surname, o_fname, o_bname, o_btype,
        o_street, o_addr_num, o_t_k, o_city);
}

```

Εάν δεν υπάρχει στη βάση εμφανίζεται “Owner not found! Please insert the following attributes:”

Στην εισαγωγή συμβολαίου ελέγχεται εάν υπάρχει η επιχείρηση και αν είναι ήδη νοικιασμένη.

Εάν υπάρχει και δεν είναι νοικιασμένη το πρόγραμμα ζητάει τα υπόλοιπα στοιχεία του συμβολαίου από τον χρήστη ελέγχοντας με ίδιο τρόπο όπως στην εισαγωγή ακινήτου εάν υπάρχει ο ενοικιαστής. Αν δεν υπάρχει εισάγεται καινούριος.

Στις περιπτώσεις ενημέρωσης των τριών οντοτήτων αφού ελεγχθεί αν υπάρχει το κλειδί που θα εισάγει ο χρήστης, ζητείται έπειτα η μεταβλητή που θα αναβαθμιστεί και η νέα τιμή της. Στην περίπτωση που η μεταβλητή είναι κάποιο foreign key το πρόγραμμα φροντίζει να την αναβαθμίσει γενικά.

Πχ στην εισαγωγή ακινήτου ένα προσπαθήσουμε να αλλάξουμε τον ιδιοκτήτη, ελέγχεται εάν υπάρχει ο ιδιοκτήτης στη βάση και αν δεν τον εισάγει.

```

MySQLConnect.java Main.java X
value2 = sc.nextInt();
conn.updateProperty(p_num1, variable, value2, null, null);
} else if (variable.equals("owner_recognize_number")) { // if we update owner_recognize_number
    // we must update owner entity as well
    value2 = sc.nextInt();
    rs2 = conn.select("owner", "recognize_number", value2);
    if (rs2 != null) { // if owner already exists upgrade
        conn.updateProperty(p_num1, variable, value2, null, null);
    } else { // else insert new one
        System.out.println("Owner doesn't exist! Please enter the following attributes:\n");
        System.out.println("Please enter owner's name or null if it is a business:\n");
        String o_name = sc.next(); // enter owner name
        System.out.println("Please enter owner's surname or null if it is a business:\n");
        String o_surname = sc.next(); // enter owner surname
        System.out.println("Please enter owner's fathername or null if it is a business:\n");
        String o_fname = sc.next(); // enter owner father name
        System.out.println("Please enter owner's business name or null if it is a person:\n");
        String o_bname = sc.next(); // enter business name
        System.out.println("Please enter owner's business type or null if it is a person:\n");
        String o_btype = sc.next(); // enter business type
        System.out.println("Please enter owner's address street (instead of space use _):\n");
        String o_street = sc.next(); // enter street from address
        System.out.println("Please enter owner's address number:\n");
        int o_addr_num = sc.nextInt(); // enter owner's address number
        System.out.println("Please enter owner's address T.K:\n");
        int o_t_k = sc.nextInt(); // enter owner's address T.K: taxydomikos kodikas
        System.out.println("Please enter owner's city of residence:\n");
        String o_city = sc.next(); // enter city of residence
        System.out.println("Please enter owner's email:\n");
        String o_email = sc.next(); // enter email
        System.out.println("Please enter owner's telephone number:\n");
        int o_tel = sc.nextInt(); // enter owner's telephone number
    }
}

```

Κάθε φορά που το πρόγραμμα εκτελεί κάποια αναβάθμιση ή εμφανίζει στον χρήστη ότι δεν υπάρχουν οι τιμές που εισήγαγε, τον ρωτάει μετά εάν θέλει να συνεχίσει την αναβάθμιση της συγκεκριμένης οντότητας και να εισάγει «1» εάν η επιλογή του είναι «Ναι» και «2» για επιλογή «Όχι».

2. Με την εισαγωγή του «2» στο αρχικό menu επιστρέφονται τα αποτελέσματα του 2<sup>ου</sup> ερωτήματος της εκφώνησης.
3. Με την εισαγωγή του «3» στο αρχικό menu το πρόγραμμα ζητάει την σημερινή ημερομηνία από τον χρήστη και επιστρέφει τα αποτελέσματα του 3<sup>ου</sup> ερωτήματος της εκφώνησης.
4. Με την εισαγωγή του «4» στο αρχικό menu το πρόγραμμα ζητάει την επιθυμητή πόλη, διεύθυνση, τετραγωνικά μέτρα, αριθμό συγκεκριμένου υποκαταστήματος και τύπο (προαιρετικό) από τον χρήστη και επιστρέφει τα αποτελέσματα του 4<sup>ου</sup> ερωτήματος της εκφώνησης.

```
647 - case 4: //checked!!
648 -> {
649 -> System.out.println("Please enter property's desired city:");
650 -> String city1 = sc.next(); //user enters property's city!!
651 -> System.out.println("Please enter the street from property's address (instead of space use _):");
652 -> String strt = sc.next(); //user enters the street from property's address!!
653 -> System.out.println("Please enter the number from property's address:");
654 -> int number = sc.nextInt(); //user enters the number from property's address!!
655 -> System.out.println("Please enter the T_K from property's address:");
656 -> int t_k = sc.nextInt(); //user enters property's T_K!!
657 -> System.out.println("Please enter property's type(or null if you don't have any preference):");
658 -> String type1 = sc.next(); //user enters property's type!!
659 -> System.out.println("Please enter the shop's number:");
660 -> int num = sc.nextInt(); //user enters shop's number!!
661 -> System.out.println("Please enter the property's square meters or:");
662 -> int sq = sc.nextInt(); //user enters square meters!!
663 -> {
664 -> conn.case4(city1, strt, number, t_k, type1, num, sq);
665 -> }
666 -> }
667 -> break;
```

5. Με την εισαγωγή του «5» στο αρχικό menu το πρόγραμμα εκτελεί εισαγωγή νέου συμβολαίου όπως στην περίπτωση 1 → 3 του προγράμματος.
6. Με την εισαγωγή του «6» στο αρχικό menu το πρόγραμμα επιστρέφει όλα τα μη νοικιασμένα ακίνητα.
7. Με την εισαγωγή του «7» στο αρχικό menu το πρόγραμμα διαγράφει τα νοικιασμένα ακίνητα.
8. Με την εισαγωγή του «8» στο αρχικό menu το πρόγραμμα επιστρέφει τις απαντήσεις στο ερώτημα 8 της εκφώνησης.
9. Με την εισαγωγή του «9» στο αρχικό menu το πρόγραμμα εμφανίζει το παρακάτω menu:

```
9
Which variable you want to upgrade?
1.email
2.telephone number
3.Address(city)
4.Address(street)
5.Address(number)
6.Address(t_k)
7.Exit
<
```

Στο οποίο εισάγει από 1 – 7 αναλόγως το τι θέλει να αναβαθμίσει. Σε κάθε περίπτωση το πρόγραμμα τον ρωτάει ποια οντότητα θέλει να αναβαθμίσει και ελέγχει εάν η μεταβλητή υπάρχει σε αυτή την οντότητα. Πχ σε περίπτωση που θέλει να αναβαθμίσει email θα πρέπει

να εισάγει “tenant”, “worker”, “owner” ή “shop” , καθώς μόνο οι συγκεκριμένες οντότητες έχουν email.

- **Γενικές Πληροφορίες:**

Η διαχείριση της βάσης δεδομένων γίνεται με τη χρήση των παρακάτω μεθόδων οι οποίες ανήκουν στην κλάση “MysqlConnect.java” :

1. insertTenant(int, String, String, String, String, String, int, int, String)
2. insertOwner(int, String, String, String, String, String, String, int, int, String)
3. insertWorker(int, String, String, String, String, String, int, String, int, int, String, int)
4. insertEmail(String, int, int, int, int)
5. insertTelephone(int, int, int, int, int)
6. insertShop(int, String, int, int, String)
7. insertProperty(int, int, String, String, int, int, String, int, int)
8. insertRental(int, String, String, String, int, int)
9. updateTenant(int, String, int, String, String)
10. updateProperty(int, String, int, String, String)
11. updateRental(int, String, int, String, String)
12. select(String, String, int)
13. case2()
14. case3(String)
15. case4(String, String, int, int, String, int, int)
16. case6()
17. case7()
18. case8()
19. updateAddress(String, String, int, String, int, String, String)
20. updateET(String, String, int, String, int, String)

Σε καθεμία από τις μεθόδους γίνεται σύνδεση με τη βάση και μετά αποσύνδεση όταν εκτελεστεί η επιθυμητή λειτουργία.

Το όνομα της βάσης που χρησιμοποιήσα είναι “project\_loans” με user: “root” και κωδικό : “ “. Εάν επιθυμείτε να χρησιμοποιήσετε βάση με άλλο όνομα μπορείτε να τοποθετήσετε τα στοιχεία της βάσης σας στις γραμμές 5, 6, 7 της κλάσης MysqlConnect.java για να συνδέεται το πρόγραμμα. (φωτογραφία κάτω)

```
se IDE
Run Window Help
MySQLConnect.java Main.java
1
2 import java.sql.*;
3
4
5 public class MySQLConnect {
6     » static String connection = "jdbc:mysql://localhost:3306/project_loans";
7     » static String user = "root";
8     » static String password = "";
9     » public static void main(String[] args) throws SQLException {
10    »     System.out.println("MySQL Connect Example");
11    »
12    »     String driver = "com.mysql.jdbc.Driver";
13    »     try {
14    »         »
15    »         »     Class.forName(driver).getDeclaredConstructor().newInstance();
16    »         »     Connection conn = DriverManager.getConnection(connection, user, password);
17    »         »     System.out.println("Connected to database");
18    »     } catch (Exception e) {
19    »         »     e.printStackTrace();
20    »     }
21    »
22 }
23
```