

Βάσεις Δεδομένων
Εξαμηνιαία Εργασία
e-Hotels

Ομάδα 103

Δημήτρης Κελέσης (03115037 - el15037@mail.ntua.gr)

Μάριος Παπαχρήστου (03115101 - papachristoumarios@gmail.com)

Παναγιώτης Τσαπατσάρης (03115102 - el15102@mail.ntua.gr)

Ακ. Έτος 2017-2018 - Ροή Α

Περιγραφή Εργασίας

Η παρούσα εξαμηνιαία εργασία αναφέρεται στη δημιουργία μιας εφαρμογής με ΒΔ για την υποστήριξη λειτουργιών e-booking για ξενοδοχειακές μονάδες σε Python 3 με χρήση του Flask web framework και της MySQL. Η εφαρμογή που να επιτρέπει σε πελάτες να κάνουν εύκολα και αποδοτικά κρατήσεις σε δωμάτια βλέποντας την διαθεσιμότητα σε πραγματικό χρόνο και επιτρέποντας στις μονάδες να έχουν μεγαλύτερα ποσοστά πληρότητας όλο τον χρόνο. Για την ορθή λειτουργία της εφαρμογής απαιτείται η ανάπτυξη μίας ΒΔ σε MySQL που θα περιέχει τις απαιτούμενες πληροφορίες ώστε να είναι δυνατές οι ενέργειες που περιγράφονται παραπάνω.

Συγκεκριμένα η εφαρμογή επιτρέπει τις εξής (κύριες) λειτουργίες:

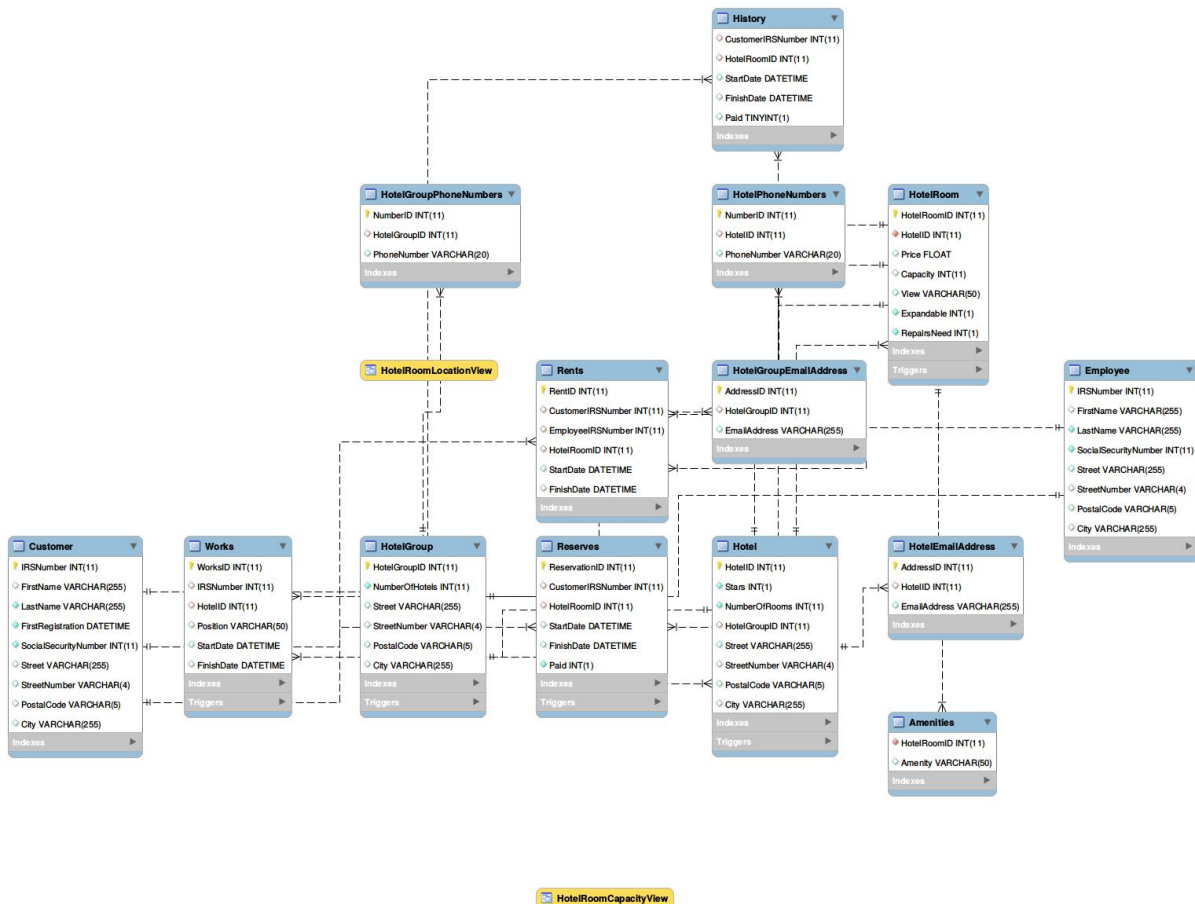
1. Να μπορούμε να επεξεργαστούμε το πελατολόγιο και τους εργαζόμενους των ξενοδοχείων, δηλαδή να υποστηρίζουμε εγγραφές πελατών / απασχολούμενων σε ξενοδοχεία, επεξεργασία των δεδομένων τους καθώς και διαγραφή τους.
2. Ο κάθε πελάτης να μπορεί να δει τα διαθέσιμα προς ενοικίαση δωμάτια με βάση πολλαπλά κριτήρια, να επιλέξει το δωμάτιο που τον ενδιαφέρει και να κάνει την αντίστοιχη κράτηση. Τα κριτήρια αυτά θα πρέπει να είναι η ημερομηνία που τον ενδιαφέρει, η χωρητικότητα του δωματίου, η περιοχή, η αλυσίδα ξενοδοχείων που ανήκει το ξενοδοχείο, η κατηγορία του ξενοδοχείου, ο συνολικός αριθμός των δωματίων που έχει το ξενοδοχείο (με εύρος τιμής), καθώς και οι παροχές του δωματίου. Τα κριτήρια αυτά θα πρέπει να είναι ανεξάρτητα, να μην απαιτούνται όλα και η λίστα που θα βλέπει ο χρήστης να ενημερώνεται με κάθε διαφοροποίηση στην επιλογή. Θα πρέπει ακόμα ο χρήστης να μπορεί να δει δύο όψεις, μία με τον αριθμό των διαθέσιμων δωματίων ανά περιοχή και μία με την χωρητικότητα όλων των δωματίων.

1. Σχεσιακό Διάγραμμα Βάσης Δεδομένων

Το δοσμένο ER διάγραμμα μεταφράζεται σε σχεσιακό διάγραμμα με τις εξής ενέργειες. Απεικονίζουμε το σχεσιακό διάγραμμα που αντιστοιχεί στο ER με τη βοήθεια του MySQL Workbench:

1. Κάθε strong entity αντιστοιχεί σε ένα πίνακα, όπως επίσης και τα Composite Attributes. Αυτό σημαίνει ότι για τις οντότητες Employee, Customer, Hotel_Group, Customer θα έχουμε τους πίνακες:
2. Κάθε M:N (ή N:K:M) σχέση αντιστοιχεί και αυτή σε ένα πίνακα επομένως έχουμε για τις σχέσεις Works, Reserves και Rents θα μας δώσουν και αυτές πίνακες:
3. Τα weak entities θα μας δώσουν πίνακες με συνδυασμό κλειδιών χωρίς (προαιρετικά primary key) αλλά με τη χρήση του unique constraint.

Το σχεσιακό διάγραμμα απεικονίζεται πλήρως παρακάτω:



2. Σχεδιασμός Βάσης Δεδομένων

2.1 Πίνακες της Βάσης και Περιορισμοί - Database Tables and Constraints

Πίνακας για τους Εργαζόμενους.

```
CREATE TABLE `Employee` (
  `IRSNumber` int(11) NOT NULL,
  `FirstName` varchar(255) DEFAULT NULL,
  `LastName` varchar(255) NOT NULL DEFAULT '',
  `SocialSecurityNumber` int(11) NOT NULL DEFAULT '0',
  `Street` varchar(255) DEFAULT NULL,
  `StreetNumber` varchar(4) DEFAULT NULL,
  `PostalCode` varchar(5) DEFAULT NULL,
  `City` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`IRSNumber`),
  KEY `IRSNumber` (`IRSNumber`,`LastName`,`FirstName`)
)
```

Ο κάθε εργαζόμενος έχει Firstname και Lastname. Ακόμα έχει SSN που δεν γίνεται να είναι 0 και αν μείνει ασυμπλήρωτο παίρνει τιμή 0. Ακόμα έχει πεδία για δρόμο, αριθμό δρόμου και ταχυδρομικό κώδικα. Ακόμα ως primary key ορίζεται το IRS.

Πίνακας για τους πελάτες.

```
CREATE TABLE `Customer` (
  `IRSNumber` int(11) NOT NULL,
  `FirstName` varchar(255) DEFAULT NULL,
  `LastName` varchar(255) NOT NULL DEFAULT '',
  `FirstRegistration` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `SocialSecurityNumber` int(11) NOT NULL DEFAULT '0',
  `Street` varchar(255) DEFAULT NULL,
  `StreetNumber` varchar(4) DEFAULT NULL,
  `PostalCode` varchar(5) DEFAULT NULL,
  `City` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`IRSNumber`),
  KEY `IRSNumber` (`IRSNumber`,`LastName`,`FirstName`)
)
```

Ο κάθε πελάτης έχει FirstName, LastName, την ημερομηνία που πρωτοεγγράφηκε στο site και τον SSN του. Ακόμα υπάρχουν πεδία για τον δρόμο, τον αριθμό και τον ταχυδρομικό κώδικα του. Όπως και πριν ως primary key ορίζεται το IRS.

Πίνακας για τις ξενοδοχειακές ομάδες.

```
CREATE TABLE `HotelGroup` (
  `HotelGroupID` int(11) NOT NULL AUTO_INCREMENT,
  `NumberOfHotels` int(11) NOT NULL DEFAULT '0',
  `Street` varchar(255) DEFAULT NULL,
  `StreetNumber` varchar(4) DEFAULT NULL,
  `PostalCode` varchar(5) DEFAULT NULL,
  `City` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`HotelGroupID`)
)
```

Το κάθε group έχει σαν πεδία το ID του που είναι και primary key, τον αριθμό ξενοδοχείων που έχει, μια διεύθυνση που αποτελείται από το όνομα του δρόμου, τον αριθμό και τον ταχυδρομικό κώδικα

Πίνακας για τα e-mail των group

```
CREATE TABLE `HotelGroupEmailAddress` (
  `AddressID` int(11) NOT NULL AUTO_INCREMENT,
  `HotelGroupID` int(11) DEFAULT NULL,
  `EmailAddress` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`AddressID`),
  KEY `HotelGroupID` (`HotelGroupID`),
  CONSTRAINT `HotelGroupEmailAddress_ibfk_1` FOREIGN KEY
  (`HotelGroupID`) REFERENCES `HotelGroup` (`HotelGroupID`)
)
```

Εδώ σαν πεδία υπάρχουν το e-mail, το AddressID που είναι το primary key, το HotelGroupID που είναι foreign key και κάνει την αντιστοίχιση με τον Πίνακα HotelGroupID

Πίνακας για τα τηλέφωνα των group.

```
CREATE TABLE `HotelGroupPhoneNumbers` (
  `NumberID` int(11) NOT NULL AUTO_INCREMENT,
  `HotelGroupID` int(11) DEFAULT NULL,
  `PhoneNumber` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`NumberID`),
```

```
KEY `HotelGroupID` (`HotelGroupID`),
CONSTRAINT `HotelGroupPhoneNumbers_ibfk_1` FOREIGN KEY
(`HotelGroupID`) REFERENCES `HotelGroup` (`HotelGroupID`)
)
```

Όπως και παραπάνω σαν πεδία υπάρχουν το NumberID που λειτουργεί ως primary key, το HotelGroupID που λειτουργεί σαν foreign key και το Phone Number που είναι ο αριθμός τηλεφώνου του κάθε ξενοδοχείου.

Πίνακας για τα ξενοδοχεία.

```
CREATE TABLE `Hotel` (
  `HotelID` int(11) NOT NULL AUTO_INCREMENT,
  `Stars` int(1) NOT NULL DEFAULT '0',
  `NumberOfRooms` int(11) NOT NULL DEFAULT '0',
  `HotelGroupID` int(11) DEFAULT NULL,
  `Street` varchar(255) DEFAULT NULL,
  `StreetNumber` varchar(4) DEFAULT NULL,
  `PostalCode` varchar(5) DEFAULT NULL,
  `City` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`HotelID`),
  KEY `HotelGroupID` (`HotelGroupID`),
  CONSTRAINT `Hotel_ibfk_1` FOREIGN KEY (`HotelGroupID`) REFERENCES
`HotelGroup` (`HotelGroupID`)
)
```

Περιλαμβάνονται το ID του κάθε ξενοδοχείου, τον αριθμό αστεριών του, τον αριθμό των δωματίων που έχει, το ID του group στο οποίο ανήκει, την διεύθυνση του που αποτελείται από τον δρόμο τον αριθμό και τον ταχυδρομικό κώδικα καθώς και το όνομα της πόλης που βρίσκεται. Σαν primary key χρησιμοποιείται το HotelID.

Πίνακας με τα e-mail των ξενοδοχείων

```
CREATE TABLE `HotelEmailAddress` (
  `AddressID` int(11) NOT NULL AUTO_INCREMENT,
  `HotelID` int(11) DEFAULT NULL,
  `EmailAddress` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`AddressID`),
  KEY `HotelID` (`HotelID`),
  CONSTRAINT `HotelEmailAddress_ibfk_1` FOREIGN KEY (`HotelID`)
REFERENCES `Hotel` (`HotelID`)
)
```

Εδώ υπάρχουν το AddressID που είναι το primary key του table, το HotelID που είναι foreign key και το EmailAddress του κάθε ξενοδοχείου.

Πίνακας για τα τηλέφωνα των ξενοδοχείων

```
CREATE TABLE `HotelPhoneNumbers` (
  `NumberID` int(11) NOT NULL AUTO_INCREMENT,
  `HotelID` int(11) DEFAULT NULL,
  `PhoneNumber` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`NumberID`),
  KEY `HotelID` (`HotelID`),
  CONSTRAINT `HotelPhoneNumbers_ibfk_1` FOREIGN KEY (`HotelID`)
REFERENCES `Hotel` (`HotelID`)
)
```

Όπως και πριν υπάρχει το NumberID που είναι primary key, το HotelID που είναι foreign key και το Phone Number.

Πίνακας για τα δωμάτια των ξενοδοχείων

```
CREATE TABLE `HotelRoom` (
  `HotelRoomID` int(11) NOT NULL AUTO_INCREMENT,
  `HotelID` int(11) NOT NULL,
  `Price` float DEFAULT NULL,
  `Capacity` int(11) DEFAULT NULL,
  `View` varchar(50) DEFAULT NULL,
  `Expandable` int(1) NOT NULL DEFAULT '0',
  `RepairsNeed` int(1) NOT NULL DEFAULT '0',
  PRIMARY KEY (`HotelRoomID`),
  UNIQUE KEY `HotelRoomID` (`HotelRoomID`,`HotelID`),
  KEY `HotelID` (`HotelID`,`HotelRoomID`),
  CONSTRAINT `HotelRoom_ibfk_1` FOREIGN KEY (`HotelID`) REFERENCES
`Hotel` (`HotelID`)
)
```

Εδώ υπάρχουν το HotelRoomID που είναι το primary key, το HotelID δηλαδή το ID του κάθε ξενοδοχείου στο οποίο ανήκει το δωμάτιο που είναι και foreign key, το Price που είναι η τιμή του δωματίου για την βραδιά, το capacity που είναι η χωρητικότητα του δωματίου, το

Expandable που είναι δυαδικό και δείχνει αν είναι επεκτάσιμο, το Repairs Need που είναι και αυτό δυαδικό και δείχνει αν χρειάζεται επισκευές ή όχι.

Πίνακας για το ιστορικό κρατήσεων

```
CREATE TABLE `History` (
  `CustomerIRSNumber` int(11) DEFAULT NULL,
  `HotelRoomID` int(11) DEFAULT NULL,
  `StartDate` datetime DEFAULT NULL,
  `FinishDate` datetime DEFAULT NULL,
  `Paid` tinyint(1) DEFAULT NULL,
  UNIQUE KEY `CustomerIRSNumber`
  (`CustomerIRSNumber`,`HotelRoomID`,`StartDate`,`FinishDate`),
  KEY `HotelRoomID` (`HotelRoomID`),
  CONSTRAINT `History_ibfk_1` FOREIGN KEY (`CustomerIRSNumber`)
  REFERENCES `Customer` (`IRSNumber`) ON DELETE SET NULL ON UPDATE
  CASCADE,
  CONSTRAINT `History_ibfk_2` FOREIGN KEY (`HotelRoomID`) REFERENCES
  `HotelRoom` (`HotelRoomID`) ON DELETE SET NULL ON UPDATE CASCADE
)
```

Σε αυτόν τον πίνακα τα πεδία είναι το IRS του πελάτη το id του δωματίου, η μέρα άφιξης, η μέρα αναχώρησης και το αν έχει πληρωθεί ή όχι.

Πίνακας για τις ανέσεις

```
CREATE TABLE `Amenities` (
  `HotelRoomID` int(11) NOT NULL,
  `Amenity` varchar(50) DEFAULT NULL,
  KEY `HotelRoomID` (`HotelRoomID`),
  CONSTRAINT `Amenities_ibfk_1` FOREIGN KEY (`HotelRoomID`) REFERENCES
  `HotelRoom` (`HotelRoomID`)
)
```

Εδώ τα πεδία είναι το id του δωματίου που είναι και foreign key και το amenity που είναι η παροχή του κάθε δωματίου

Πίνακας για τις κρατήσεις

```
CREATE TABLE `Reserves` (
  `ReservationID` int(11) NOT NULL AUTO_INCREMENT,
```



```
`CustomerIRSNumber` int(11) DEFAULT NULL,
`HotelRoomID` int(11) DEFAULT NULL,
`StartDate` datetime DEFAULT NULL,
`FinishDate` datetime DEFAULT NULL,
`Paid` int(1) NOT NULL DEFAULT '0',
PRIMARY KEY (`ReservationID`),
KEY `CustomerIRSNumber` (`CustomerIRSNumber`),
KEY `HotelRoomID` (`HotelRoomID`),
CONSTRAINT `Reserves_ibfk_1` FOREIGN KEY (`CustomerIRSNumber`)
REFERENCES `Customer` (`IRSNumber`) ON DELETE SET NULL ON UPDATE
CASCADE,
CONSTRAINT `Reserves_ibfk_2` FOREIGN KEY (`HotelRoomID`) REFERENCES
`HotelRoom` (`HotelRoomID`) ON DELETE SET NULL ON UPDATE CASCADE
)
```

Τα πεδία όπως φαίνονται είναι το ReservationID που είναι το primary key του πίνακα, το CustomerIRSNumber και το HotelRoomID που είναι foreign keys, η ημερομηνία άφιξης και αναχώρησης και το αν έχει πληρωθεί ή όχι.

Πίνακας για τις ενοικιάσεις

```
CREATE TABLE `Rents` (
  `RentID` int(11) NOT NULL AUTO_INCREMENT,
  `CustomerIRSNumber` int(11) DEFAULT NULL,
  `EmployeeIRSNumber` int(11) DEFAULT NULL,
  `HotelRoomID` int(11) DEFAULT NULL,
  `StartDate` datetime DEFAULT NULL,
  `FinishDate` datetime DEFAULT NULL,
  PRIMARY KEY (`RentID`),
  KEY `CustomerIRSNumber` (`CustomerIRSNumber`),
  KEY `EmployeeIRSNumber` (`EmployeeIRSNumber`),
  KEY `HotelRoomID` (`HotelRoomID`),
  CONSTRAINT `Rents_ibfk_1` FOREIGN KEY (`CustomerIRSNumber`) REFERENCES
`Customer` (`IRSNumber`) ON DELETE SET NULL ON UPDATE CASCADE,
  CONSTRAINT `Rents_ibfk_2` FOREIGN KEY (`EmployeeIRSNumber`) REFERENCES
`Employee` (`IRSNumber`) ON DELETE SET NULL ON UPDATE CASCADE,
  CONSTRAINT `Rents_ibfk_3` FOREIGN KEY (`HotelRoomID`) REFERENCES
`HotelRoom` (`HotelRoomID`) ON DELETE SET NULL ON UPDATE CASCADE
)
```

Στο παραπάνω table τα πεδία είναι το RentID που είναι το primary key μαλιστα,το customerIRSNumber και το EmployeeIRSNumber που είναι foreign keys καθώς και το

HotelRoomID το οποίο και αυτό είναι foreign key και τέλος οι ημερομηνίες άφιξης και αναχώρησης.

Πίνακας για τις σχέσεις εργαζόμενων-ξενοδοχείων

```
CREATE TABLE `Works` (
  `WorksID` int(11) NOT NULL AUTO_INCREMENT,
  `IRSNumber` int(11) DEFAULT NULL,
  `HotelID` int(11) DEFAULT NULL,
  `Position` varchar(50) DEFAULT NULL,
  `StartDate` datetime DEFAULT NULL,
  `FinishDate` datetime DEFAULT NULL,
  PRIMARY KEY (`WorksID`),
  KEY `IRSNumber` (`IRSNumber`),
  KEY `HotelID` (`HotelID`),
  CONSTRAINT `Works_ibfk_1` FOREIGN KEY (`IRSNumber`) REFERENCES
`Employee` (`IRSNumber`),
  CONSTRAINT `Works_ibfk_2` FOREIGN KEY (`HotelID`) REFERENCES `Hotel`
(`HotelID`)
)
```

Τα πεδία εδώ είναι το WorksID που είναι primary key, το IRSNumber που είναι foreign key όπως και το HotelID του ξενοδοχείου στο οποίο αυτός εργάζεται, το Position που υποδηλώνει την θέση του εργαζομένου και τέλος το από πότε ξεκίνησε να δουλεύει καθώς και την ημερομηνία όποτε σταμάτησε αν αυτή υπάρχει.

2.2 Περιορισμοί Βάσης και Διακόπτες - Database Constraints & Triggers

Οι triggers της βάσης βρίσκονται στο αρχείο sql/triggers.sql και είναι οι ακόλουθοι:
use eHOTELS;

```
delimiter $$
create trigger delroomgr
before delete on Hotel
for each row
begin
update HotelGroup set NumberOfHotels = NumberOfHotels - 1 where
HotelGroupID = old.HotelGroupID;

end; $$
```

```
create trigger delroom
before delete on HotelRoom
for each row
begin
update Hotel set NumberOfRooms = NumberOfRooms - 1 where HotelRoomID =
old.HotelRoomID;

end; $$
```

```
delimiter $$
create trigger reservtrig1
after delete on Reserves
for each row
begin
set @x=(SELECT IRSNumber FROM Works WHERE HotelID=(SELECT HotelID FROM
HotelRoom WHERE HotelRoomID=old.HotelRoomID) AND Position='Manager');
insert into Rents (CustomerIRSNumber, EmployeeIRSNumber, HotelRoomID,
StartDate, FinishDate)
VALUES (old.CustomerIRSNumber,@x ,old.HotelRoomID, old.StartDate,
old.FinishDate);
end; $$

delimiter $$
create trigger nonoverlap
before insert on Reserves
for each row
begin
IF EXISTS (select * from Reserves
```

```

        where (HotelRoomID = new.HotelRoomID)
        and ( (new.StartDate between StartDate and FinishDate) or
              (new.FinishDate between StartDate and FinishDate) or
              (StartDate between new.StartDate and new.FinishDate) or
              (FinishDate between new.StartDate and new.FinishDate))
    ) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Overlap found!';
    END IF;

end; $$

delimiter $$
create trigger history
after insert on Reserves
for each row
begin
    insert into History (CustomerIRSNumber, HotelRoomID, StartDate,
FinishDate, Paid)
        values (new.CustomerIRSNumber, new.HotelRoomID, new.StartDate,
new.FinishDate, new.Paid);
end; $$

delimiter $$
create trigger deletemgr
before delete on Works
for each row
begin
    IF EXISTS (select * from Works where HotelID = old.HotelID and Position
= 'Manager'
    ) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'No manager exists for this hotel!';
    END IF;

end; $$

delimiter $$
create trigger updatemgr
before update on Works
for each row
begin

```

```

IF EXISTS (select * from Works where HotelID = old.HotelID and Position
= 'Manager'
) THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'No manager exists for this hotel!';
END IF;

end; $$

delimiter $$
create trigger update_nr_hotels
before insert on Hotel
for each row
begin
    update HotelGroup set NumberOfHotels = NumberOfHotels + 1
    where HotelGroupID = new.HotelGroupID;
end; $$

delimiter $$
create trigger update_nr_rooms
before insert on HotelRoom
for each row
begin
    update Hotel set NumberOfRooms = NumberOfRooms + 1
    where HotelID = new.HotelID;
end; $$

drop trigger rentcheck;
delimiter $$
create trigger rentcheck
before insert on Rents
for each row
begin
IF not EXISTS (
    select * from WorksHotelRoom where WorksHotelRoom.IRSNumber =
new.EmployeeIRSNumber
    and WorksHotelRoom.HotelRoomID = new.HotelRoomID
) THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Wrong Employee!';
END IF;

end; $$

```

```

end; $$

delimiter $$
create trigger chkprice
before insert on HotelRoom
for each row
begin
IF (new.Price < 0) THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Negative Price!';
END IF;

end; $$

delimiter $$
create trigger chknrrooms
before insert on Hotel
for each row
begin
IF (new.NumberOfRooms < 0) THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Negative number of rooms!';
END IF;

end; $$

delimiter $$
create trigger chknrhotels
before insert on HotelGroup
for each row
begin
IF (new.NumberOfHotels < 0) THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Negative number of hotels!';
END IF;

end; $$

delimiter $$
create trigger chknrhotels
before insert on HotelGroup
for each row
begin
IF (new.NumberOfHotels < 0) THEN
    SIGNAL SQLSTATE '45000'

```

```

    SET MESSAGE_TEXT = 'Negative number of hotels!';
END IF;

end; $$

delimiter $$
create trigger phonecheck before insert on HotelGroupPhoneNumbers
for each row
begin
IF (new.PhoneNumber regexp '^(\\+?[0-9]{1,4}-)?[0-9]{3,10}$' ) = 0 THEN
    SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Wrong phone number!';
end if;
END$$

delimiter $$
create trigger chkstreet
before insert on Employee
for each row
begin
IF (new.StreetNumber < 0) THEN
    SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Negative street number!';
END IF;

end; $$

delimiter $$
create trigger chkstreet2
before insert on Customer
for each row
begin
IF (new.StreetNumber < 0) THEN
    SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Negative street number!';
END IF;

end; $$

delimiter $$
create trigger chkstreet3
before insert on Hotel
for each row
begin
IF (new.StreetNumber < 0) THEN

```

```

    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Negative street number!';
END IF;

end; $$

delimiter $$
create trigger chkstars
before insert on Hotel
for each row
begin
IF not(new.Stars between 0 and 5) THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'No valid stars!';
END IF;

end; $$

```

2.2 Ευρετήρια

Customer (IRSNumber, FirstName, LastName)

Πρωτεύοντα στοιχεία στην αναζήτηση πελατών είναι το ονοματεπώνυμο τους και το αφμ τους

Employee(IRSNumber, FirstName, LastName)

Όπως και στους πελάτες έτσι και στους εργαζόμενους οι αναζητήσεις γίνονται με βάση το ΑΦΜ τους και το ονοματεπώνυμο τους

Works(WorksID, IRSNumber, HotelID)

Στην σχέση works τα πεδία με τα οποία θα αναζητήσουμε κάτι θα είναι κατά πάσα πιθανότητα το ΑΦΜ του εργαζομένου, το ξενοδοχείο στο οποίο δουλεύει και κυρίως το primary key της σχέσης

HotelGroup(HotelGroupID)

Εδώ μας απασχολεί μόνο το ID και χρησιμοποιούμε το default της mysql

Hotel(HotelID,HotelGroupID)

Τα πεδία εδώ είναι το HotelGroupID και το primary key το HotelID

HotelRoom(HotelRoomID, HotelID)

Το δωμάτιο αναζητείται με βάση το ID του και το ID του ξενοδοχείου στο οποίο ανήκει.

Reserves(ReservationID, CustomerID, RoomID, StartDate, FinishDate)

Rents(RentID, CustomerID, RoomID, StartDate, FinishDate)

Αυτό που μας ενδιαφέρει σε μία κράτηση και σε μία ενοικίαση αντίστοιχα είναι το ID της, το ΑΦΜ του πελάτη που την κάνει, το ID του δωματίου για το οποίο γίνεται και οι ημερομηνίες έναρξης και λήξης. Θα μπορούσαμε εναλλακτικά να βάλουμε σαν κύριο ευρετήριο τις ημ/νιες έναρξης & λήξης δεδομένου ότι οι κρατήσεις συνήθως ψάχνονται κατ' ημερομηνία σε πραγματικές συνθήκες.

Για τους υπόλοιπους πίνακες επειδή χρησιμοποιούμε τα default indexes της MySQL.

3. Τεχνολογία Ανάπτυξης Εφαρμογής

Η εφαρμογή αναπτύχθηκε σε Python 3 με χρήση του MySQL driver. Επιλέξαμε αυτή τη λύση διότι είναι προγραμματιστικά εύκολη και μας προσφέρει μεγάλη ευελιξία. Για τη διαχείριση της βάσης χρησιμοποιήθηκε το MySQL Workbench. Για το γραφικό περιβάλλον (frontend) έχουμε χρησιμοποιήσει HTML 5 και CSS για το αισθητικό κομμάτι. Η εφαρμογή αποτελείται από σελίδες (views) οι οποίες χειρίζονται GET και POST requests. Συγκεκριμένα οι σελίδες που φορτώνονται προκειμένου να δείξουν στοιχεία υλοποιούν GET ενώ αυτές που διαχειρίζονται δεδομένα φορμών διαχειρίζονται μεθόδους POST.¹

Οι πληροφορίες περνάνε μέσα από τις φόρμες οι οποίες γίνονται submit (POST request) και εν συνεχεία αναλαμβάνει η Python για να τρέξει τα queries που χρειάζονται στη βάση και να άλλες λειτουργίες του backend. Εκεί γίνεται επίσης χειρισμός εξαιρέσεων από την Python για διάφορες εξαιρέσεις και την εμφάνιση καταλλήλων μηνυμάτων λάθους στο χρήστη.

Ο connector της βάσης αρχικοποιείται με την έναρξη του application, συνδέεται στη βάση και μετά από αυτό δημιουργούμε έναν cursor ο οποίος είναι υπεύθυνος για το τρέξιμο των queries και την λήψη πληροφοριών (fetch) από τη βάση σε περίπτωση που κάνουμε ερωτήσεις (π.χ. select) ή commit σε περίπτωση που επηρεάζουμε τα περιεχόμενα της βάσης. Για να διασφαλιστεί η απαιτούμενη ασφάλεια από λανθασμένες ενέργειες υπάρχουν έλεγχοι τόσο στο frontend (στις φόρμες), στο backend (conditions) όσο και στην ίδια τη βάση με triggers.

¹ REST API

4. Εγκατάσταση Εφαρμογής

Ο οδηγός εγκατάστασης που ακολουθεί αναφέρεται σε συστήματα GNU/Linux σε διανομές Ubuntu και Debian. Η διαδικασία που ακολουθείται είναι αντίστοιχη και για άλλα συστήματα.

4.1 Εγκατάσταση Συστήματος Βάσης Δεδομένων - MySQL Server

Εγκαθιστούμε τον MySQL Server στο σύστημα μας με τη βοήθεια των εντολών

```
sudo apt-get update && sudo apt-get install mysql-server &&
mysql_secure_installation
```

Στην παραπάνω διαδικασία θα ερωτηθούμε να θέσουμε κάποιους κωδικούς για να έχουμε πρόσβαση στον MySQL server.

Για να βεβαιωθούμε ότι ο MySQL server τρέχει σωστά μπορούμε να ελέγξουμε με

```
systemctl status mysql.service
```

Αν θέλουμε να τον ξεκινήσουμε

```
sudo service mysql start
```

Μπορούμε τέλος να κάνουμε login στον MySQL server με την ακόλουθη εντολή

```
mysql -u {username} -p
```

Αφού βεβαιωθούμε ότι ο MySQL server δουλεύει σωστά, μπορούμε να προχωρήσουμε στην διαδικασία δημιουργίας της βάσης.

4.2 Δημιουργία Βάσης για υποστήριξη του DBMS

Κάνουμε login στον MySQL server και δημιουργούμε τη βάση

```
Create database eHOTELS;
Use eHOTELS;
```

Στην εφαρμογή παρέχονται κάποια SQL scripts στη διαδρομή sql/ για την δημιουργία των απαραίτητων πινάκων. Συγκεκριμένα υπάρχουν scripts για τη δημιουργία πινάκων (create_tables.sql), triggers (create_triggers.sql) και εισαγωγή δεδομένων (hotel_seeder.sql).

Έχοντας κάνει cd στο directory του project και login στον MySQL server μπορούμε να τρέξουμε αυτά τα scripts με την εντολή

```
Source sql/create_tables.sql;
Source sql/create_triggers.sql;
Source sql/hotel_seeder.sql;
```

Τα παραπάνω scripts θα στήσουν τη βάση για να προχωρήσουμε στο στήσιμο του περιβάλλοντος της Python 3 και τέλος για την εκτέλεση του web application.

4.3 Εγκατάσταση Περιβάλλοντος Python 3 - MySQL Connector - Flask

Έχοντας στήσει τη βάση, προχωράμε στο στήσιμο του περιβάλλοντος πάνω στο οποίο τρέχει η εφαρμογή που έχουμε υλοποιήσει. Σε ένα τερματικό εγκαθιστούμε τα απαραίτητα πακέτα

```
sudo apt-get install python3 python3-dev python3-flask python3-pip
python3-mysql.connector
```

4.4 Ρύθμιση και Εκτέλεση Εφαρμογής

Μέσα στο root directory της εφαρμογής δημιουργούμε ένα αρχείο credentials.config με τα εξής περιεχόμενα

```
[DBCredentials]
DB=eHOTELS
USER=foo
PASSWORD=puppy
PORT=3306
HOST=localhost
```

Με τις απαιτούμενες ρυθμίσεις για τη βάση. Έχοντας ολοκληρώσει αυτό το βήμα, είμαστε έτοιμοι να εκτελέσουμε την εφαρμογή είτε κάνοντας διπλό κλικ στο run.sh είτε με την εντολή:

```
export FLASK_APP=app.py && flask run &
```

Είμαστε έτοιμοι! Ανοίγουμε τον browser μας και πλοηγούμαστε στη διεύθυνση

```
http://localhost:5000
```

Εγκατάσταση σε Windows

Εγκατάσταση από .exe αρχείο που γίνεται download από την παρακάτω διεύθυνση

https://repo.anaconda.com/archive/Anaconda3-5.1.0-Windows-x86_64.exe

Αφού εγκαταστήσουμε το anaconda στον υπολογιστή μας ανοίγουμε το anaconda command prompt και πληκτρολογούμε

```
pip install Flask
```

Αφού έχουμε εγκαταστήσει το Flask πρέπει να εγκαταστήσουμε το mysql server που γίνεται από την παρακάτω διεύθυνση.

<https://dev.mysql.com/get/Downloads/MySQLInstaller/mysql-installer-community-8.0.11.0.msi>

Έπειτα για να γίνει επιτυχής η εγκατάσταση του mysql connector στο Anaconda command prompt πληκτρολογούμε:

```
pip install mysql-connector-python-rf
```

Και αφού έχουμε κάνει cd στο directory που υπάρχει το app.py εκτελούμε την εντολή

```
python app.py
```

Και η εφαρμογή ανοίγει στον browser.

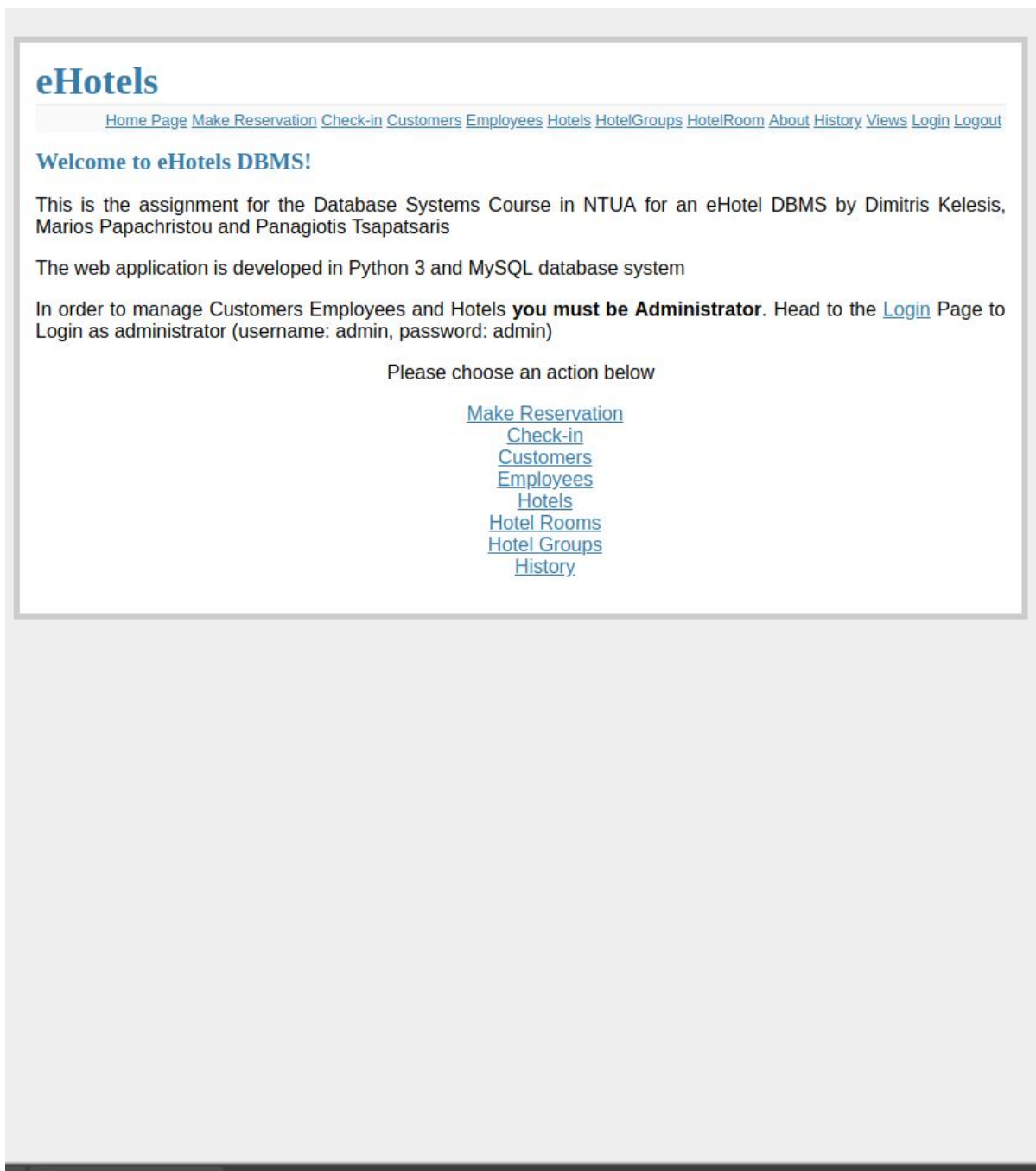
Κώδικας Κατασκευής Βάσης

Το DDL που κατασκευάζει όλη τη βάση μαζί με τα στοιχεία βρίσκεται στο αρχείο sql/db.sql και είναι ένα dump της βάσης που έχει δημιουργηθεί με το εργαλείο mysqldump. Η εκτέλεση αυτού του SQL script θα δημιουργήσει τη βάση με όλα τα απαραίτητα στοιχεία.

5. Ενδεικτικές Σελίδες Εφαρμογής

Θα δούμε πως κάνουμε κράτηση για ένα δωμάτιο από την βάση μας

Αυτή είναι η αρχική σελίδα:



Από αυτήν την σελίδα μπορούμε να κάνουμε μια κράτηση(Make Reservation) να κάνουμε check-in και με την προϋπόθεση πως έχουμε κάνει login ως administrator μπορούμε να δούμε τους εργαζόμενους,τους πελάτες, τα ξενοδοχεία, τα δωμάτια,τα group και το ιστορικό των κρατήσεων.

Ας δούμε τώρα πως γίνεται να κάνουμε μία κράτηση. Αφού κάνουμε κλικ πάνω στον σύνδεσμο Make Reservation. Εμφανίζεται μπροστά μας αυτή η άδεια φόρμα

eHotels

[Home Page](#)
[Make Reservation](#)
[Check-in](#)
[Customers](#)
[Employees](#)
[Hotels](#)
[HotelGroups](#)
[HotelRoom](#)
[About](#)
[History Views](#)
[Login](#)
[Logout](#)

Make a reservation

Please select your preferences below

Minimum Price (min 0)

Maximum Price

Minimum Stars (min 0)

Maximum Stars (max 5, default 5)

Check-in Date

Check-out Date

View

Number of bed

City

Hotel Group ID

Number Of Available Rooms

Amenities

☐ AC
☐ 2WC
☐ Massage
☐ Sauna

Ordered by

☐ Capacity
☐ Region

Search

Αφού συμπληρώσουμε κάποιες τιμές σε αυτήν γίνεται κάπως έτσι:

eHotels

[Home Page](#)
[Make Reservation](#)
[Check-in](#)
[Customers](#)
[Employees](#)
[Hotels](#)
[HotelGroups](#)
[HotelRoom](#)
[About](#)
[History](#)
[Views](#)
[Login](#)
[Logout](#)

Make a reservation

Please select your preferences below

Minimum Price (min 0)

Maximum Price

Minimum Stars (min 0)

Maximum Stars (max 5, default 5)

Check-in Date

Check-out Date

View

Number of bed

City

Hotel Group ID

Number Of Available Rooms

Amenities

☐ AC
☐ 2WC
☐ Massage
☐ Sauna

Ordered by

☐ Capacity
☒ Region

Και πατάμε το search button για να γίνει η αναζήτηση. Προκύπτει το ακόλουθο αποτέλεσμα:

Success! Here are your results for searching.

| | |
|---------------|------------|
| MinimumPrice | 0 |
| MinimumStars | 0 |
| MaximumStars | 5 |
| StartDate | 2018-07-12 |
| FinishDate | 2018-07-16 |
| Capacity | 3 |
| City | Athens |
| NumberOfRooms | 3 |
| orderBy | City |

Ordered by :

| | |
|--------|-------|
| City | Count |
| Athens | 3 |

Your search returned 3 results

| | |
|--------------|----------------------------------|
| HotelRoomID | 3 |
| HotelID | 1 |
| Price | 94.0 |
| Capacity | 3 |
| View | Sea |
| Stars | 2 |
| City | Athens |
| Street | Papastratou |
| StreetNumber | 10 |
| PostalCode | 17237 |
| | Make Reservation |

| | |
|--------------|----------------------------------|
| HotelRoomID | 8 |
| HotelID | 2 |
| Price | 299.0 |
| Capacity | 3 |
| View | City |
| Stars | 3 |
| City | Athens |
| Street | Imittou |
| StreetNumber | 3 |
| PostalCode | 41231 |
| | Make Reservation |

| | |
|--------------|----------------------------------|
| HotelRoomID | 53 |
| HotelID | 11 |
| Price | 125.0 |
| Capacity | 3 |
| View | Mountain |
| Stars | 1 |
| City | Athens |
| Street | Ilioupolis |
| StreetNumber | 12 |
| PostalCode | 16263 |
| | Make Reservation |

Πατώντας Make Reservation στο πρώτο ερχόμαστε σε αυτήν την οθόνη την οποία συμπληρώνουμε:

eHotels

[Home Page](#) [Make Reservation](#) [Check-in](#) [Customers](#) [Employees](#) [Hotels](#) [HotelGroups](#) [HotelRoom](#) [About](#) [History](#) [Views](#) [Login](#) [Logout](#)

Make a reservation

Please put in your IRS Number

First Name
Panagiotis

Last Name
Tsapatsaris

Make Reservation

Και αφού πατήσουμε Make Reservation παίρνουμε το IRSNumber μας.

eHotels

[Home Page](#) [Make Reservation](#) [Check-in](#) [Customers](#) [Employees](#) [Hotels](#) [HotelGroups](#) [HotelRoom](#) [About](#) [History](#) [Views](#) [Login](#) [Logout](#)

IRSNumber : 13213
First Name: Panagiotis
Last Name: Tsapatsaris

Next

Και πατώντας Next επιβεβαιώνουμε την κράτηση μας.

eHotels

[Home Page](#) [Make Reservation](#) [Check-in](#) [Customers](#) [Employees](#) [Hotels](#) [HotelGroups](#) [HotelRoom](#) [About](#) [History](#) [Views](#) [Login](#) [Logout](#)

Your reservation was successfull!

Your reservation ID is 2

Looking forward to seeing you soon :)

Αφού έχουμε κάνει κράτηση πρέπει να κάνει check in ο υπάλληλος ο οποίος έχει αναλάβει την κράτηση από αυτό το παράθυρο

eHotels

[Home Page](#) [Make Reservation](#) [Check-in](#) [Customers](#) [Employees](#) [Hotels](#) [HotelGroups](#) [HotelRoom](#) [About](#) [History](#) [Views](#) [Login](#) [Logout](#)

Manage Checkins

This is the Checkin Page

Checkin Details

Reservation ID
2

Employee IRS Number

Payment Method
CREDIT

Check in

Τέλος εδώ φαίνεται το ιστορικό των κρατήσεων.

eHotels

[Home Page](#) [Make Reservation](#) [Check-in](#) [Customers](#) [Employees](#) [Hotels](#) [HotelGroups](#) [HotelRoom](#) [About](#) [History](#) [Views](#) [Login](#) [Logout](#)

History

Here are last reservations

| | |
|-------------------|---------------------|
| HotelRoomID | 1 |
| StartDate | 1973-01-01 00:00:01 |
| Paid | 0 |
| CustomerIRSNumber | 3 |
| FinishDate | 1974-01-01 00:00:01 |

| | |
|-------------------|---------------------|
| HotelRoomID | 12 |
| StartDate | 2018-05-22 19:44:48 |
| Paid | 0 |
| CustomerIRSNumber | 123214 |
| FinishDate | 2018-05-22 19:44:48 |

| | |
|-------------------|---------------------|
| HotelRoomID | 12 |
| StartDate | 2018-05-22 19:44:54 |
| Paid | 0 |
| CustomerIRSNumber | 123214 |
| FinishDate | 2018-05-22 19:44:54 |

| | |
|-------------------|---------------------|
| HotelRoomID | 12 |
| StartDate | 2018-05-22 19:45:15 |
| Paid | 0 |
| CustomerIRSNumber | 123214 |
| FinishDate | 2018-05-22 19:45:15 |

Τώρα θα δούμε ενδεικτικά κάποιες δυνατότητες που έχουμε.

eHotels

[Home Page](#)
[Make Reservation](#)
[Check-in](#)
[Customers](#)
[Employees](#)
[Hotels](#)
[HotelGroups](#)
[HotelRoom](#)
[About](#)
[History](#)
[Views](#)
[Login](#)
[Logout](#)

Manage Customers

This is the Customers Page. From here you can create a customer, find a customer and alter his details or delete existing customers

Customer Details

IRS Number

First Name

Last Name

Social Security Number

Address (Street)

Number

Postal Code

City

You can create a customer from here [Create a customer](#)

Από εδώ μπορούμε να βρούμε κάποιους customers καθώς και να δημιουργήσουμε νέους.

eHotels

[Home Page](#)
[Make Reservation](#)
[Check-in](#)
[Customers](#)
[Employees](#)
[Hotels](#)
[HotelGroups](#)
[HotelRoom](#)
[About](#)
[History](#)
[Views](#)
[Login](#)
[Logout](#)

Manage Employees

This is the Employees Page. From here you can create an employee, find an employee and alter his details or delete existing employees

Employee Details

IRS Number

First Name

Last Name

Social Security Number

Address (Street)

Number

Postal Code

City

Search

You can create a employee from here [Create an Employee](#)

Όπως και πριν αλλά για εργαζόμενους. Και παρακάτω για ξενοδοχεία

eHotels

[Home Page](#)
[Make Reservation](#)
[Check-in](#)
[Customers](#)
[Employees](#)
[Hotels](#)
[HotelGroups](#)
[HotelRoom](#)
[About](#)
[History](#)
[Views](#)
[Login](#)
[Logout](#)

Manage Hotels

This is the Hotels Page. From here you can create an employee, find a hotel and alter his details or delete existing hotels

Hotels Details

Hotel ID

Stars

Hotel Group ID

Number of Rooms

Address (Street)

Number

Postal Code

City

Search

You can create a hotel from here [Create a hotel](#)

Τα παραπάνω μπορούμε να τα κάνουμε για Γκρουπ ξενοδοχείων καθώς και για τα δωμάτια.

Σημειώσεις

Για version control του κώδικα χρησιμοποιήθηκε git από τα μέλη της ομάδας. Τα συστήματα ανάπτυξης ήταν Ubuntu GNU/Linux 16.04 LTS. Για το χειρισμό της βάσης χρησιμοποιήθηκε το πρόγραμμα MySQL Workbench.

Αναφορές

[1] Silberschatz, Abraham, Henry F. Korth, and Shashank Sudarshan. Database system concepts. Vol. 4. New York: McGraw-Hill, 1997.

[2] MySQL Reference Manual

[3] Σημειώσεις Μαθήματος