



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ
ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ**

ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

ΕΞΑΜΗΝΙΑΙΟ PROJECT

ΑΝΑΦΟΡΑ

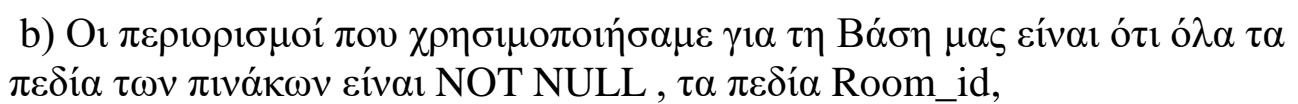
ΟΜΑΔΑ 136

Δήμητρα Πίντζου 03115181

Βαγγέλης Αποστολάκης 03114164

Στέλιος Τσαγκαράκης 03115180

α. Για τον σχεδιασμό της Βάσης Δεδομένων χρησιμοποιήσαμε το παρακάτω σχεσιακό μοντέλο :



Reservation_Number, Renting_Number, Amenity_id των πινάκων Hotel_Room, reserves, rents, Amenities αντίστοιχα είναι auto increment. Επίσης ορισμένα από τα foreign keys που χρησιμοποιήσαμε έχουν τον περιορισμό CASCADE που σημαίνει ότι όταν κάνουμε update ή delete, οι αλλαγές που κάνουμε σε ένα αντικείμενο στο οποίο αναφέρονται και άλλα αντικείμενα επηρεάζουν και αυτά τα ίδια. Τέλος θέσαμε τον περιορισμό ότι όταν κάνεις INSERT μια καταχώρηση στην οποία απαιτείται foreign key, πρέπει αυτό το foreign key να έχει ήδη εισαχθεί στη βάση.

c) Τα ευρετήρια που δημιουργήσαμε είναι τα εξής:

```
CREATE UNIQUE INDEX `idEmployees_UNIQUE` ON  
`Basic`.`Employee` (`IRS_Number` ASC);
```

```
CREATE UNIQUE INDEX `social_security_number_UNIQUE` ON  
`Basic`.`Employee` (`Social_Security_Number` ASC);
```

```
CREATE UNIQUE INDEX `hotel_id_UNIQUE` ON  
`Basic`.`Hotel_Group` (`Hotel_Group_id` ASC);
```

```
CREATE UNIQUE INDEX `hotel_id_UNIQUE` ON `Basic`.`Hotel`  
(`Hotel_id` ASC);
```

```
CREATE INDEX `fk_Hotel_Hotel_Group1_idx` ON `Basic`.`Hotel`  
(`Hotel_Group_id` ASC);
```

```
CREATE UNIQUE INDEX `IRS_Number_UNIQUE` ON  
`Basic`.`Customer` (`IRS_Number` ASC);
```

```
CREATE UNIQUE INDEX `Room_id_UNIQUE` ON  
`Basic`.`Hotel_Room` (`Room_id` ASC);
```

```
CREATE INDEX `fk_Hotel_Room_Hotel1_idx` ON  
`Basic`.`Hotel_Room` (`Hotel_id` ASC);
```

```
CREATE INDEX `fk_works_Hotel1_idx` ON `Basic`.`works` (`Hotel_id`  
ASC);
```

```
CREATE INDEX `fk_reserves_Customer1_idx` ON `Basic`.`reserves`  
(`Customer_IRS_Number` ASC);
```

```
CREATE UNIQUE INDEX `Reservation_IRS_Number_UNIQUE` ON  
`Basic`.`reserves` (`Reservation_Number` ASC);
```

```
CREATE INDEX `fk_reserves_Hotel_Room1_idx` ON `Basic`.`reserves`  
(`Room_id` ASC, `Hotel_id` ASC);
```

```
CREATE INDEX `fk_rents_Customer1_idx` ON `Basic`.`rents`  
(`Customer_IRS_Number` ASC);
```

```
CREATE UNIQUE INDEX `Renting_IRS_Number_UNIQUE` ON  
`Basic`.`rents` (`Renting_Number` ASC);
```

```
CREATE INDEX `fk_rents_Hotel_Room1_idx` ON `Basic`.`rents`  
(`Room_id` ASC, `Hotel_id` ASC);
```

```
CREATE INDEX `fk_Payment_Transaction_rents1_idx` ON  
`Basic`.`Payment_Transaction` (`Renting_Number` ASC);
```

```
CREATE INDEX `fk_Hotel_Email_Address_Hotel1_idx` ON  
`Basic`.`Hotel_Email_Address` (`Hotel_id` ASC);
```

```
CREATE UNIQUE INDEX `Email_UNIQUE` ON  
`Basic`.`Hotel_Email_Address` (`Email` ASC);
```

```
CREATE UNIQUE INDEX `Phone_UNIQUE` ON  
`Basic`.`Hotel_Phone_Number` (`Phone` ASC);
```

```
CREATE UNIQUE INDEX `Email_UNIQUE` ON  
`Basic`.`Hotel_Group_Email_Address` (`Email` ASC);
```

```
CREATE UNIQUE INDEX `Phone_UNIQUE` ON  
`Basic`.`Hotel_Group_Phone_Number` (`Phone` ASC);
```

```
CREATE UNIQUE INDEX `Amenity_id_UNIQUE` ON  
`Basic`.`Amenities` (`Amenity_id` ASC);
```

Ο λόγος που χρησιμοποιήσαμε κάποια ευρετήρια, είναι γιατί αυτά επιταχύνουν σε μεγάλο βαθμό την εκτέλεση των Queries που

χρησιμοποιούν τα εν λόγω γνωρίσματα. Για το λόγο αυτό, χρησιμοποιήσαμε ευρετήρια κυρίως για τα Primary και Foreign keys, μιας και αυτά χρησιμοποιούνται κατά κόρον.

d) Για την ανάπτυξη της εφαρμογής χρησιμοποιήσαμε το σύστημα apache-web και τις γλώσσες προγραμματισμού PHP, Html και SQL.

e) Για να εγκαταστήσει κάποιος την εφαρμογή μας , πρέπει αρχικά να δημιουργήσει μια βάση δεδομένων όπως αυτή περιγράφεται από τον κώδικα SQL που παραθέτουμε στο ερώτημα (3). Έπειτα, πρέπει να γεμίσει τη βάση (με δεδομένα της επιλογής του). Τέλος πρέπει να συνδέσει με τη χρήση των αρχείων PHP που δίνονται στο ερώτημα (4) την βάση αυτή με ένα περιβάλλον φιλικό για το χρήστη, το οποίο περιγράφεται από τα αρχεία HTML που δίνονται επίσης στο ερώτημα (4).

2. Τα DDL που κατασκευάζουν τη βάση φαίνονται παρακάτω :

Όψεις/Views :

Χρησιμοποιήσαμε τις δύο παρακάτω όψεις με τις οποίες εμφανίζονται τα δωμάτια ανά πόλη και η χωρητικότητα των δωματίων. Οι όψεις αυτές είναι πολύ χρήσιμες αφού αποκρύπτονται πληροφορίες που ίσως να μην ενδιαφέρουν τον χρήστη όπως η διεύθυνση του ξενοδοχείου, τα αστέρια του, οι παροχές των δωματίων κ.λ.π. Αυτό διευκολύνει να παρατηρήσει κανείς ανά πάσα στιγμή πόσα δωμάτια υπάρχουν σε κάθε πόλη ή την χωρητικότητα του κάθε δωματίου.

```
-- -----  
-- View `Basic`.`Rooms_per_City`  
-- -----
```

```
DROP TABLE IF EXISTS `Basic`.`Rooms_per_City`;  
USE `Basic`;  
CREATE OR REPLACE VIEW `Rooms_per_City` AS select  
`Hotel`.`City` AS `City`,  
SUM(`Hotel`.`Number_of_Rooms`) AS `Rooms`  
from `Hotel`  
group by `Hotel`.`City`;  
-- -----
```

```
-- View `Basic`.`Room_Capacity`
-----
DROP TABLE IF EXISTS `Basic`.`Room_Capacity`;
USE `Basic`;
CREATE OR REPLACE VIEW `Room_Capacity` AS select
`Hotel_Room`.`Hotel_id` AS `Hotel`,
`Hotel_Room`.`Room_id` AS `Room_Number`,
`Hotel_Room`.`Capacity` AS `Capacity`
from `Hotel_Room`
order by `Hotel_Room`.`Capacity` ASC;
USE `Basic`;
```

Triggers :

Χρησιμοποιήσαμε τα παρακάτω triggers στη βάση μας :

```
CREATE DEFINER = CURRENT_USER TRIGGER
`Basic`.`New_Hotel` AFTER INSERT ON `Hotel` FOR EACH ROW
BEGIN
    UPDATE Hotel_Group hg
    SET hg.Number_of_Hotels = (SELECT COUNT(*) FROM Hotel
WHERE (Hotel_Group_id = hg.Hotel_Group_id));
END
```

```
CREATE DEFINER = CURRENT_USER TRIGGER
`Basic`.`Change_Hotel` AFTER UPDATE ON `Hotel` FOR EACH ROW
BEGIN
    UPDATE Hotel_Group hg
    SET hg.Number_of_Hotels = (SELECT COUNT(*) FROM Hotel
WHERE (Hotel_Group_id = hg.Hotel_Group_id));
END
```

```
CREATE DEFINER = CURRENT_USER TRIGGER
`Basic`.`Close_Hotel` AFTER DELETE ON `Hotel` FOR EACH ROW
BEGIN
    UPDATE Hotel_Group hg
    SET hg.Number_of_Hotels = (SELECT COUNT(*) FROM Hotel
WHERE (Hotel_Group_id = hg.Hotel_Group_id));
```

END

```
CREATE DEFINER = CURRENT_USER TRIGGER
`Basic`.`New_Hotel_Room` AFTER INSERT ON `Hotel_Room` FOR
EACH ROW
BEGIN
    UPDATE Hotel h
    SET h.Number_of_Rooms = (SELECT COUNT(*) FROM
Hotel_Room WHERE (Hotel_id = h.Hotel_id));
END
```

```
CREATE DEFINER = CURRENT_USER TRIGGER
`Basic`.`Change_Hotel_Room` AFTER UPDATE ON `Hotel_Room` FOR
EACH ROW
BEGIN
    UPDATE Hotel h
    SET h.Number_of_Rooms = (SELECT COUNT(*) FROM
Hotel_Room WHERE (Hotel_id = h.Hotel_id));
END
```

```
CREATE DEFINER = CURRENT_USER TRIGGER
`Basic`.`Close_Hotel_Room` AFTER DELETE ON `Hotel_Room` FOR
EACH ROW
BEGIN
    UPDATE Hotel h
    SET h.Number_of_Rooms = (SELECT COUNT(*) FROM
Hotel_Room WHERE (Hotel_id = h.Hotel_id));
END
```

```
CREATE DEFINER = CURRENT_USER TRIGGER `Basic`.`Paid`
AFTER INSERT ON `rents` FOR EACH ROW
BEGIN
    UPDATE reserves r
    SET r.Paid = 1
    WHERE r.Room_id = NEW.Room_id AND
r.Customer_IRS_Number = NEW.Customer_IRS_Number AND
r.Start_Date = NEW.Start_Date;
END
```

Τα πρώτα 3 triggers αφορούν τα ξενοδοχεία (Hotel στον πίνακα της βάσης) και ο ρόλος τους είναι να ανανεώνουν τον αριθμό ξενοδοχείων του κάθε Hotel_group όταν έχουμε διαγραφή, εκχώρηση νέου ή ανανέωση στοιχείων ενός ξενοδοχείου. Αντιστοίχως έχουμε 3 triggers και για τα δωμάτια ξενοδοχείου (Hotel_Room στον πίνακα της βάσης) για να ανανεώνουν τον αριθμό των δωματίων που ανήκουν σε κάθε ξενοδοχείο για διαγραφή, εκχώρηση ή ανανέωση όπως πριν. Τέλος έχουμε ένα τελευταίο trigger για την ενοικίαση (rents) η οποία όταν γίνεται ενημερώνει το πεδίο paid μετατρέποντας το σε 1 , το οποίο βρίσκεται στις κρατήσεις (reserves).

3. Ο κώδικας SQL που γράψαμε για να υποστηρίξουμε τη βάση φαίνεται παρακάτω :

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS,
UNIQUE_CHECKS=0;
SET
@OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';
```

```
-- -----
-- Schema Basic
-- -----
```

```
-- -----
-- Schema Basic
-- -----
```

```
CREATE SCHEMA IF NOT EXISTS `Basic` DEFAULT CHARACTER
SET utf8 ;
USE `Basic` ;
```

```
-- -----
-- Table `Basic`.`Employee`
-- -----
```

```
CREATE TABLE IF NOT EXISTS `Basic`.`Employee` (
  `IRS_Number` INT NOT NULL,
  `Social_Security_Number` INT NOT NULL,
```



```
`Last_Name` VARCHAR(45) NOT NULL,  
`First_Name` VARCHAR(45) NOT NULL,  
`Street` VARCHAR(45) NOT NULL,  
`Number` INT NOT NULL,  
`City` VARCHAR(45) NOT NULL,  
`Postal_Code` INT NOT NULL,  
PRIMARY KEY (`IRS_Number`))  
ENGINE = InnoDB;
```

```
CREATE UNIQUE INDEX `idEmployees_UNIQUE` ON  
`Basic`.`Employee` (`IRS_Number` ASC);
```

```
CREATE UNIQUE INDEX `social_security_number_UNIQUE` ON  
`Basic`.`Employee` (`Social_Security_Number` ASC);
```

```
-- -----  
-- Table `Basic`.`Hotel_Group`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `Basic`.`Hotel_Group` (  
  `Hotel_Group_id` VARCHAR(45) NOT NULL,  
  `Number_of_Hotels` SMALLINT NOT NULL DEFAULT 0,  
  `Street` VARCHAR(45) NOT NULL,  
  `Number` INT NOT NULL,  
  `Postal_Code` INT NOT NULL,  
  `City` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`Hotel_Group_id`))  
ENGINE = InnoDB;
```

```
CREATE UNIQUE INDEX `hotel_id_UNIQUE` ON  
`Basic`.`Hotel_Group` (`Hotel_Group_id` ASC);
```

```
-- -----  
-- Table `Basic`.`Hotel`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `Basic`.`Hotel` (  
  `Hotel_id` VARCHAR(45) NOT NULL,  
  `Number_of_Rooms` SMALLINT NOT NULL DEFAULT 0,
```

```

`Stars` TINYINT(5) NOT NULL,
`Street` VARCHAR(45) NOT NULL,
`Number` INT NOT NULL,
`Postal_Code` INT NOT NULL,
`City` VARCHAR(45) NOT NULL,
`Hotel_Group_id` VARCHAR(45) NOT NULL,
PRIMARY KEY (`Hotel_id`),
CONSTRAINT `fk_Hotel_Hotel_Group1`
  FOREIGN KEY (`Hotel_Group_id`)
  REFERENCES `Basic`.`Hotel_Group` (`Hotel_Group_id`)
  ON DELETE CASCADE
  ON UPDATE CASCADE)
ENGINE = InnoDB;

```

```

CREATE UNIQUE INDEX `hotel_id_UNIQUE` ON `Basic`.`Hotel`
(`Hotel_id` ASC);

```

```

CREATE INDEX `fk_Hotel_Hotel_Group1_idx` ON `Basic`.`Hotel`
(`Hotel_Group_id` ASC);

```

```

-- -----
-- Table `Basic`.`Customer`
-- -----

```

```

CREATE TABLE IF NOT EXISTS `Basic`.`Customer` (
  `IRS_Number` INT NOT NULL,
  `Last_Name` VARCHAR(45) NOT NULL,
  `First_Name` VARCHAR(45) NOT NULL,
  `Social_Security_Number` INT NOT NULL,
  `First_Registration` DATE NOT NULL,
  `Street` VARCHAR(45) NOT NULL,
  `City` VARCHAR(45) NOT NULL,
  `Number` INT NOT NULL,
  `Postal_Code` INT NOT NULL,
  PRIMARY KEY (`IRS_Number`))
ENGINE = InnoDB;

```

```

CREATE UNIQUE INDEX `IRS_Number_UNIQUE` ON
`Basic`.`Customer` (`IRS_Number` ASC);

```

-- Table `Basic`.`Hotel_Room`

```
CREATE TABLE IF NOT EXISTS `Basic`.`Hotel_Room` (  
  `Room_id` SMALLINT NOT NULL AUTO_INCREMENT,  
  `Capacity` TINYINT NOT NULL,  
  `View` VARCHAR(45) NOT NULL,  
  `Expandable` TINYINT NOT NULL,  
  `Repairs_Need` TINYINT(1) NOT NULL,  
  `Price` FLOAT NOT NULL,  
  `Hotel_id` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`Room_id`, `Hotel_id`),  
  CONSTRAINT `fk_Hotel_Room_Hotel1`  
    FOREIGN KEY (`Hotel_id`)  
      REFERENCES `Basic`.`Hotel` (`Hotel_id`)  
      ON DELETE CASCADE  
      ON UPDATE CASCADE)  
ENGINE = InnoDB;
```

```
CREATE UNIQUE INDEX `Room_id_UNIQUE` ON  
`Basic`.`Hotel_Room` (`Room_id` ASC);
```

```
CREATE INDEX `fk_Hotel_Room_Hotel1_idx` ON  
`Basic`.`Hotel_Room` (`Hotel_id` ASC);
```

-- Table `Basic`.`works`

```
CREATE TABLE IF NOT EXISTS `Basic`.`works` (  
  `Start_Date` DATE NOT NULL,  
  `Finish_Date` DATE NOT NULL,  
  `Position` VARCHAR(45) NOT NULL DEFAULT "",  
  `Employee_IRS_Number` INT NOT NULL,  
  `Hotel_id` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`Employee_IRS_Number`),  
  CONSTRAINT `fk_works_Employee`
```

```

FOREIGN KEY (`Employee_IRS_Number`)
REFERENCES `Basic`.`Employee` (`IRS_Number`)
ON DELETE CASCADE
ON UPDATE CASCADE,
CONSTRAINT `fk_works_Hotel1`
FOREIGN KEY (`Hotel_id`)
REFERENCES `Basic`.`Hotel` (`Hotel_id`)
ON DELETE CASCADE
ON UPDATE CASCADE)
ENGINE = InnoDB;

CREATE INDEX `fk_works_Hotel1_idx` ON `Basic`.`works` (`Hotel_id`
ASC);

```

```

-----
-- Table `Basic`.`reserves`
-----

```

```

CREATE TABLE IF NOT EXISTS `Basic`.`reserves` (
  `Start_Date` DATE NOT NULL,
  `Finish_Date` DATE NOT NULL,
  `Paid` TINYINT(1) NOT NULL DEFAULT 0,
  `Customer_IRS_Number` INT NOT NULL,
  `Reservation_Number` INT NOT NULL AUTO_INCREMENT,
  `Room_id` SMALLINT NOT NULL,
  `Hotel_id` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`Reservation_Number`),
  CONSTRAINT `fk_reserves_Customer1`
    FOREIGN KEY (`Customer_IRS_Number`)
    REFERENCES `Basic`.`Customer` (`IRS_Number`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_reserves_Hotel_Room1`
    FOREIGN KEY (`Room_id` , `Hotel_id`)
    REFERENCES `Basic`.`Hotel_Room` (`Room_id` , `Hotel_id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;

```

```
CREATE INDEX `fk_reserves_Customer1_idx` ON `Basic`.`reserves`  
(`Customer_IRS_Number` ASC);
```

```
CREATE UNIQUE INDEX `Reservation_IRS_Number_UNIQUE` ON  
`Basic`.`reserves` (`Reservation_Number` ASC);
```

```
CREATE INDEX `fk_reserves_Hotel_Room1_idx` ON `Basic`.`reserves`  
(`Room_id` ASC, `Hotel_id` ASC);
```

```
-----  
-- Table `Basic`.`rents`  
-----
```

```
CREATE TABLE IF NOT EXISTS `Basic`.`rents` (  
  `Start_Date` DATE NOT NULL,  
  `Finish_Date` DATE NOT NULL,  
  `Employee_IRS_Number` INT NOT NULL,  
  `Customer_IRS_Number` INT NOT NULL,  
  `Renting_Number` INT NOT NULL AUTO_INCREMENT DEFAULT 0,  
  `Room_id` SMALLINT NOT NULL,  
  `Hotel_id` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`Renting_Number`),  
  CONSTRAINT `fk_rents_Employee1`  
    FOREIGN KEY (`Employee_IRS_Number`)  
    REFERENCES `Basic`.`Employee` (`IRS_Number`)  
    ON DELETE NO ACTION  
    ON UPDATE CASCADE,  
  CONSTRAINT `fk_rents_Customer1`  
    FOREIGN KEY (`Customer_IRS_Number`)  
    REFERENCES `Basic`.`Customer` (`IRS_Number`)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE,  
  CONSTRAINT `fk_rents_Hotel_Room1`  
    FOREIGN KEY (`Room_id` , `Hotel_id`)  
    REFERENCES `Basic`.`Hotel_Room` (`Room_id` , `Hotel_id`)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE)  
ENGINE = InnoDB;
```

```
CREATE INDEX `fk_rents_Customer1_idx` ON `Basic`.`rents`  
(`Customer_IRS_Number` ASC);
```

```
CREATE UNIQUE INDEX `Renting_IRS_Number_UNIQUE` ON  
`Basic`.`rents` (`Renting_Number` ASC);
```

```
CREATE INDEX `fk_rents_Hotel_Room1_idx` ON `Basic`.`rents`  
(`Room_id` ASC, `Hotel_id` ASC);
```

```
-----  
-- Table `Basic`.`Payment_Transaction`  
-----
```

```
CREATE TABLE IF NOT EXISTS `Basic`.`Payment_Transaction` (  
  `Payment_Amount` FLOAT NOT NULL,  
  `Payment_Method` VARCHAR(45) NOT NULL,  
  `Renting_Number` INT NOT NULL,  
  PRIMARY KEY (`Renting_Number`),  
  CONSTRAINT `fk_Payment_Transaction_rents1`  
    FOREIGN KEY (`Renting_Number`)  
    REFERENCES `Basic`.`rents` (`Renting_Number`)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE)  
ENGINE = InnoDB;
```

```
CREATE INDEX `fk_Payment_Transaction_rents1_idx` ON  
`Basic`.`Payment_Transaction` (`Renting_Number` ASC);
```

```
-----  
-- Table `Basic`.`Hotel_Email_Address`  
-----
```

```
CREATE TABLE IF NOT EXISTS `Basic`.`Hotel_Email_Address` (  
  `Hotel_id` VARCHAR(45) NOT NULL,  
  `Email` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`Email`),  
  CONSTRAINT `fk_Hotel_Email_Address_Hotel1`  
    FOREIGN KEY (`Hotel_id`)  
    REFERENCES `Basic`.`Hotel` (`Hotel_id`)
```

```
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;
```

```
CREATE INDEX `fk_Hotel_Email_Address_Hotel1_idx` ON
`Basic`.`Hotel_Email_Address` (`Hotel_id` ASC);
```

```
CREATE UNIQUE INDEX `Email_UNIQUE` ON
`Basic`.`Hotel_Email_Address` (`Email` ASC);
```

```
-----
-- Table `Basic`.`Hotel_Phone_Number`
-----
```

```
CREATE TABLE IF NOT EXISTS `Basic`.`Hotel_Phone_Number` (
  `Phone` BIGINT NOT NULL,
  `Hotel_id` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`Phone`),
  CONSTRAINT `fk_Hotel_Phone_Number_Hotel1`
    FOREIGN KEY (`Hotel_id`)
    REFERENCES `Basic`.`Hotel` (`Hotel_id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;
```

```
CREATE UNIQUE INDEX `Phone_UNIQUE` ON
`Basic`.`Hotel_Phone_Number` (`Phone` ASC);
```

```
-----
-- Table `Basic`.`Hotel_Group_Email_Address`
-----
```

```
CREATE TABLE IF NOT EXISTS
`Basic`.`Hotel_Group_Email_Address` (
  `Email` VARCHAR(45) NOT NULL,
  `Hotel_Group_id` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`Email`),
  CONSTRAINT `fk_Hotel_Group_Email_Address_Hotel_Group1`
    FOREIGN KEY (`Hotel_Group_id`)
```

```
REFERENCES `Basic`.`Hotel_Group` (`Hotel_Group_id`)
ON DELETE CASCADE
ON UPDATE CASCADE)
ENGINE = InnoDB;
```

```
CREATE UNIQUE INDEX `Email_UNIQUE` ON
`Basic`.`Hotel_Group_Email_Address` (`Email` ASC);
```

```
-----
-- Table `Basic`.`Hotel_Group_Phone_Number`
--
```

```
-----
CREATE TABLE IF NOT EXISTS
`Basic`.`Hotel_Group_Phone_Number` (
  `Phone` BIGINT NOT NULL,
  `Hotel_Group_id` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`Phone`),
  CONSTRAINT `fk_Hotel_Group_Phone_Number_Hotel_Group1`
  FOREIGN KEY (`Hotel_Group_id`)
  REFERENCES `Basic`.`Hotel_Group` (`Hotel_Group_id`)
  ON DELETE CASCADE
  ON UPDATE CASCADE)
ENGINE = InnoDB;
```

```
CREATE UNIQUE INDEX `Phone_UNIQUE` ON
`Basic`.`Hotel_Group_Phone_Number` (`Phone` ASC);
```

```
-----
-- Table `Basic`.`Amenities`
--
```

```
-----
CREATE TABLE IF NOT EXISTS `Basic`.`Amenities` (
  `Amenity` VARCHAR(45) NOT NULL,
  `Room_id` SMALLINT NOT NULL,
  `Amenity_id` INT NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`Amenity_id`),
  CONSTRAINT `fk_Amenities_Hotel_Room1`
  FOREIGN KEY (`Room_id`)
```



```
REFERENCES `Basic`.`Hotel_Room` (`Room_id`)
ON DELETE CASCADE
ON UPDATE CASCADE)
ENGINE = InnoDB;
```

```
CREATE UNIQUE INDEX `Amenity_id_UNIQUE` ON
`Basic`.`Amenities` (`Amenity_id` ASC);
```

```
USE `Basic` ;
```

```
-----
-- Placeholder table for view `Basic`.`Rooms_per_City`
-----
```

```
CREATE TABLE IF NOT EXISTS `Basic`.`Rooms_per_City` (`City`
INT, `Rooms_per_City` INT);
```

```
-----
-- Placeholder table for view `Basic`.`Room_Capacity`
-----
```

```
CREATE TABLE IF NOT EXISTS `Basic`.`Room_Capacity` (`Hotel`
INT, `Room_Number` INT, `Capacity` INT);
```

```
-----
-- View `Basic`.`Rooms_per_City`
-----
```

```
DROP TABLE IF EXISTS `Basic`.`Rooms_per_City`;
USE `Basic`;
CREATE OR REPLACE VIEW `Rooms_per_City` AS select
`Hotel`.`City` AS `City`,
SUM(`Hotel`.`Number_of_Rooms`) AS `Rooms`
from `Hotel`
group by `Hotel`.`City`;
```

```
INSERT INTO `Basic`.`Employee` (`IRS_Number` ,
`Social_Security_Number` , `Last_Name` , `First_Name` , `Street` ,
`Number` , `City` , `Postal_Code` ) VALUES
(1056754 , 29029758 , 'Pintzou' , 'Dimitra' ,
'Artemonos' , 109 , 'Athens' ,
13451 ) ,
```

(1568966 , 27099698 , 'Apostolakis' , 'Vaggelis' ,
 'Mpiskini' , 27 , 'Irakleio' , 58962) ,
 (1898681 , 28019798 , 'Tsagkarakis' , 'Stelakis' , 'Papagou' ,
 3 , 'Thessaloniki' , 78963) ,
 (1752967 , 27059469 , 'Karpouzias' , 'Vaggelis' , 'Iasonos
 Maratou' , 52 , 'Athens' , 14587) ,
 (1589876 , 23129379 , 'Plevris' , 'Aggelos' ,
 'Vouliagmenis' , 191 , 'Athens' , 13145) ,
 (1976871 , 22089798 , 'Afouxenidis' , 'Stathis' ,
 'Ellinorossos' , 55 , 'Athens' , 78955) ,
 (1358797 , 21059797 , 'Panagiotopoulou' , 'Vasiliki' , 'Xarilaou
 Trikoupi' , 14 , 'Athens' , 45809) ,
 (1878896 , 23119789 , 'Tourli' , 'Pagona' , 'Vasileos
 Georgiou' , 18 , 'Athens' , 13125) ,
 (1987535 , 21089786 , 'Mpoumi' , 'Zoi' ,
 'Grigoriou Lampraki' , 44 , 'Patra' , 97863) ,
 (1789658 , 27099765 , 'Filippopoulou' , 'Ioanna' , 'Agiou
 Thoma' , 8 , 'Volos' , 78968) ,
 (1135879 , 21049789 , 'Gkornik' , 'Nicole' ,
 'Alexandras' , 87 , 'Zakynthos' , 58979) ,
 (1087687 , 28129787 , 'Koutsouroupa' , 'Iris' ,
 'Agiass Lavras' , 33 , 'Rethimno' , 87896) ,
 (1354478 , 25059589 , 'Pintzou' , 'Stavriana' ,
 'Afroditis' , 4 , 'Ithaki' ,
 86535) ,
 (1489576 , 24019678 , 'Garantzioti' , 'Christina' ,
 'Merkouri' , 16 , 'Preveza' , 78581) ,
 (1791632 , 23089489 , 'Georgiou' , 'Marios' ,
 'Anastasaki' , 25 , 'Lamia' , 98868) ,
 (1887523 , 27098725 , 'Prousalidis' , 'Konstantinos' ,
 'Kolokotroni' , 88 , 'Kefalonia' , 98561) ,
 (1875634 , 21086983 , 'Sakalis' , 'Aggelos' ,
 'Agiou Konstantinou' , 72 , 'Preveza' , 54986) ,
 (1985315 , 23129765 , 'Oikonomou' , 'Georgios' ,
 'Ampelokipon' , 57 , 'Katerini' , 98657) ,
 (1246879 , 28068975 , 'Tzortzis' , 'Panagiotis' ,
 'Kousidi' , 101 , 'Mykonos' , 57597) ,
 (1576575 , 21028978 , 'Athina' , 'Touliatou' ,
 'Antheon' , 29 , 'Kavala' , 85630) ,

```

(1357854 , 23086984 , 'Markella' , 'Zorba' ,
  'Xolargou' , 21 , 'Skiathos' , 87601 ) ,
(1862368 , 29053674 , 'Apostolia' , 'Drakopoulou' ,
  'Kaniggos' , 87 , 'Chalkidiki' , 63254) ,
(1451254 , 23129875 , 'Theodora' , 'Christidou' ,
  'Agiass Aikaterinis' , 96 , 'Katerini' , 13779 ) ,
(1852543 , 28049751 , 'Katerina' , 'Panagiotou' ,
  'Zoodoxou Pigis' , 19 , 'Rethimno' , 87896 ) ,
(1399845 , 29109257 , 'Mixelis' , 'Anagnostakis' ,
  'Smirnis' , 33 , 'Chalkidiki' , 63254 ) ,
(1542956 , 27079789 , 'Giannis' , 'Tolias' ,
  'Leoforos Fylis' , 25 , 'Athens' , 68941 ) ,
(1796547 , 23098754 , 'Xristos' , 'Papadedes' ,
  'Kleious' , 7 , 'Thessaloniki' , 88743 ) ,
(1487455 , 26068765 , 'Spiridon' , 'Tsiolis' ,
  'Sokratous' , 15 , 'Patra' , 36781 ) ,
(1834120 , 21039876 , 'Nikoleta' , 'Aga' ,
  'Dodonis' , 38 , 'Kefalonia' , 97830 ) ,
(1796541 , 25078478 , 'Dimitris' , 'Heliotis' ,
  'Thesprotias' , 12 , 'Volos' , 34789 );

```

```

INSERT INTO `Basic`.`Hotel_Group` (`Hotel_Group_id` ,
`Number_of_Hotels` , `Street` , `Number` , `Postal_Code` , `City` )
VALUES
('Hilton' , 0 , 'Vasilisis Sofias' , 181 , 55684 ,
  'Athens' ) ,
('Sani Hotels' , 0 , 'Egnatia' , 236 , 65735 ,
  'Thessaloniki' ) ,
('Great Britain' , 0 , 'Megalou Aleksandrou' , 52 , 98663 ,
  'Patra' ) ,
('Hotel Palaces' , 0 , 'Troon' , 12 , 87965 ,
  'Volos' ) ,
('Resorts Paradise' , 0 , 'Agiou Gerasimou' , 72 , 98634 ,
  'Kefalonia' );

```

```

INSERT INTO `Basic`.`Hotel` (`Hotel_id` , `Number_of_Rooms` , `Stars` ,
`Street` , `Number` , `Postal_Code` , `City` , `Hotel_Group_id` ) VALUES

```

('Hilton Athens' , 0, 5 , 'Vasilisis Sofias' , 108, 15489 ,
 'Athens' , 'Hilton') ,
 ('Hilton Athena' , 0, 4 , 'Aiolou' ,
 33 , 15468 , 'Athens' , 'Hilton') ,
 ('Hilton Thessaloniki' , 0, 5 , 'Tsimiski' , 79 , 22364 ,
 'Thessaloniki' , 'Hilton') ,
 ('Hilton Patra' , 0, 3 , 'Korinthou' , 152 ,
 87898 , 'Patra' , 'Hilton') ,
 ('Hilton Mykonos' , 0, 5 , 'Agias Annas' , 28 , 62301 ,
 'Mykonos' , 'Hilton') ,

('Hotel Zeus' , 0, 4 , 'Agiou Georgiou' , 81 , 16954 ,
 'Zakynthos' , 'Resorts Paradise') ,
 ('Hotel Alexandra' , 0, 3 , 'Alexandras' , 182 , 15468 ,
 'Athens' , 'Resorts Paradise') ,
 ('Hotel Ermis' , 0, 5 , 'Makrygianni' , 43 , 36588 , 'Ithaki' ,
 'Resorts Paradise') ,
 ('Hotel Apollon' , 0, 4 , 'Dimokratias' , 25 , 58439 ,
 'Kefalonia' , 'Resorts Paradise') ,
 ('Hotel Artemis' , 0, 3 , 'Riga Fereou' , 87 , 98768 ,
 'Zakynthos' , 'Resorts Paradise') ,

('Filoxenia' , 0, 4 , 'Patision' , 190 , 56554 , 'Athens' ,
 'Hotel Palaces') ,
 ('Venus' , 0, 3 , 'Pindou' , 32 , 89566 , 'Volos' ,
 'Hotel Palaces') ,
 ('Aiolos' , 0, 4 , 'Tzavella' , 8 , 87303 , 'Katerini' , 'Hotel
 Palaces') ,
 ('Sorokos' , 0, 4 , 'Troias' , 12 , 10783 , 'Preveza' ,
 'Hotel Palaces') ,
 ('Xenofon' , 0, 5 , 'Themidos' , 12, 15690 , 'Athens' ,
 'Hotel Palaces') ,

('Hotel Stelakis' , 0, 5 , 'Ektoros' , 68 , 15468 ,
 'Patra' , 'Great Britain') ,
 ('Hotel Astoria' , 0, 5 , 'Kalokairinou' , 40 , 13547 ,
 'Thessaloniki' , 'Great Britain') ,

('Hotel El Greco', 0, 5, 'Leoforos Eleytherias', 61, 16954, 'Irakleio', 'Great Britain'),
 ('Hotel Rinis', 0, 4, 'Praxitelous', 16, 16954, 'Thessaloniki', 'Great Britain'),
 ('Hotel Aristeidis', 0, 3, 'Kekrops', 8, 16954, 'Rethimno', 'Great Britain'),

('Platon', 0, 5, 'Eleutheriou Venizelou', 59, 16954, 'Chalkidiki', 'Sani Hotels'),
 ('Aristotelis', 0, 4, 'Papaflessa', 61, 16954, 'Thessaloniki', 'Sani Hotels'),
 ('Sofoklis', 0, 4, 'Notara', 13, 16954, 'Kavala', 'Sani Hotels'),
 ('Evripidis', 0, 5, 'Nimfon', 26, 16954, 'Chalkidiki', 'Sani Hotels'),
 ('Sokratis', 0, 3, 'Dionisou', 74, 16954, 'Skiathos', 'Sani Hotels');

INSERT INTO `Basic`.`Customer` (`IRS_Number`, `Last_Name`, `First_Name`, `Social_Security_Number`, `First_Registration`, `Street`, `Number`, `City`, `Postal_Code`) VALUES

(5940152, 'Aguirre', 'Kibo', 89736510, '2012-03-18', 'Wieze Rd.', 178, 'California', 50081),
 (2569111, 'Booker', 'Britanney', 16375204, '2002-12-18', 'Opoeteren Rd.', 250, 'Washington', 19062),
 (7262104, 'Hodges', 'Maya', 78953140, '2005-09-17', 'Borgerhout Rd.', 30, 'Portland', 27227),
 (8776057, 'Trevino', 'Meredith', 13126973, '2017-08-07', 'Lowell Rd.', 514, 'Las Vegas', 40891),
 (8811489, 'Delaney', 'Evan', 13697069, '2016-08-09', 'Somma Lombardo Rd.', 28, 'Casper', 68391),
 (1431867, 'Sandoval', 'Theodore', 83987391, '2004-08-18', 'Montaldo Bormida', 386, 'Sofia', 77135),
 (9759624, 'Ochoa', 'Jaquelyn', 51493549, '2017-07-04', 'Chamborde', 387, 'Lyon', 20589),
 (5330641, 'Sawyer', 'Kieran', 43982872,

'2016-12-01' , 'Heist-op-den-Berg', 191, 'Berlin' , 84215),
 (4879142, 'Nikoleta', 'Koutsokosta' , 56877520 , '2009-
 12-17' , 'Evripidou' , 68, 'Athens', 22613),
 (7867580, 'Cotton', 'Ella', 16536973 ,
 '2017-12-02' , 'Saint Michelle', 65, 'Versailles', 19157),
 (2530550, 'Mendoza', 'Ira', 32190982 ,
 '2017-06-12' , 'Benalla' , 213, 'Valencia' ,
 19048),
 (6156699, 'Cleveland', 'Maya', 55873685 ,
 '2017-06-04' , 'Saint-Marcel', 691, 'Regina' ,
 29159),
 (6075692, 'Henderson', 'India' , 73806491 ,
 '2017-04-01' , 'Box Rd.' , 234, 'Luisiana', 17045),
 (9508141, 'Terry', 'Shelly' , 93056930 ,
 '2001-04-18' , 'Forges Rd.' , 781, 'Brighton' ,
 76689),
 (3495429, 'Pickett', 'Diana', 63419287 ,
 '2017-02-05' , 'Bedford' , 10 , 'Bristol' , 31504);

```

INSERT INTO `Basic`.`Hotel_Room` ( `Capacity` , `View` ,
`Expandable` , `Repairs_Need` , `Price` , `Hotel_id` ) VALUES
(1 , 'POOL' , 0 , 1 , 70 , 'Hilton Athens' ) ,
(2 , 'POOL' , 1 , 0 , 120 , 'Hilton Athens'),
(3 , 'POOL' , 1 , 0 , 90 , 'Hilton Athens'),
(4 , 'POOL' , 0 , 0 , 120 , 'Hilton Athens'),
(5 , 'POOL' , 0 , 1 , 140 , 'Hilton Athens') ,
(1 , 'POOL' , 1 , 0 , 80 , 'Hilton Athena' ) ,
(2 , 'POOL' , 1 , 0 , 100 , 'Hilton Athena' ),
(3 , 'POOL' , 0 , 1 , 110 , 'Hilton Athena' ) ,
(4 , 'POOL' , 0 , 0 , 120 , 'Hilton Athena' ) ,
(5 , 'POOL' , 1 , 1 , 150 , 'Hilton Athena' ) ,
(1 , 'SEA' , 1 , 0 , 75 , 'Hilton Thessaloniki' ) ,
(2 , 'SEA' , 1 , 0 , 90 , 'Hilton Thessaloniki' ) ,
(3 , 'POOL' , 1 , 0 , 100 , 'Hilton Thessaloniki' ) ,
(4 , 'POOL' , 0 , 0 , 120 , 'Hilton Thessaloniki'),
(5 , 'POOL' , 0 , 0 , 150 , 'Hilton Thessaloniki' ) ,
(1 , 'SEA' , 0 , 0 , 60 , 'Hilton Patra'),
(2 , 'SEA' , 0 , 1 , 70 , 'Hilton Patra' ) ,
(3 , 'POOL' , 1 , 0 , 80 , 'Hilton Patra' ) ,

```

(4 ,	'SEA' ,	0 ,	0 ,	90 ,	'Hilton Patra'),
(5 ,	'SEA' ,	0 ,	0 ,	110 ,	'Hilton Patra') ,
(1 ,	'SEA' ,	1 ,	0 ,	90 ,	'Hilton Mykonos') ,
(2 ,	'SEA' ,	1 ,	0 ,	110 ,	'Hilton Mykonos'),
(3 ,	'SEA' ,	1 ,	1 ,	115 ,	'Hilton Mykonos'),
(4 ,	'SEA' ,	0 ,	0 ,	125 ,	'Hilton Mykonos'),
(5 ,	'POOL' ,	0 ,	0 ,	150 ,	'Hilton Mykonos') ,
(1 ,	'SEA' ,	1 ,	0 ,	50 ,	'Hotel Zeus') ,
(2 ,	'SEA' ,	1 ,	0 ,	60 ,	'Hotel Zeus'),
(3 ,	'SEA' ,	0 ,	0 ,	70 ,	'Hotel Zeus'),
(4 ,	'SEA' ,	0 ,	1 ,	80 ,	'Hotel Zeus') ,
(5 ,	'POOL' ,	0 ,	0 ,	90 ,	'Hotel Zeus'),
(1 ,	'POOL' ,	1 ,	0 ,	45 ,	'Hotel Alexandra') ,
(2 ,	'POOL' ,	1 ,	0 ,	55 ,	'Hotel Alexandra'),
(3 ,	'POOL' ,	1 ,	1 ,	60 ,	'Hotel Alexandra'),
(4 ,	'POOL' ,	0 ,	0 ,	70 ,	'Hotel Alexandra') ,
(5 ,	'POOL' ,	0 ,			
	0 ,	80 ,			'Hotel Alexandra'),
(1 ,	'SEA' ,	1 ,	0 ,	60 ,	'Hotel Ermis'),
(2 ,	'SEA' ,	1 ,	0 ,	65 ,	'Hotel Ermis'),
(3 ,	'POOL' ,	1 ,	1 ,	70 ,	'Hotel Ermis'),
(4 ,	'SEA' ,	0 ,	0 ,	80 ,	'Hotel Ermis'),
(5 ,	'SEA' ,	0 ,	0 ,	90 ,	'Hotel Ermis'),
(1 ,	'SEA' ,	1 ,	0 ,	60 ,	'Hotel Apollon'),
(2 ,	'SEA' ,	1 ,	1 ,	70 ,	'Hotel Apollon') ,
(3 ,	'SEA' ,	0 ,	1 ,	80 ,	'Hotel Apollon') ,
(4 ,	'SEA' ,	0 ,	0 ,	90 ,	'Hotel Apollon') ,
(5 ,	'POOL' ,	0 ,	0 ,	100 ,	'Hotel Apollon') ,
(1 ,	'SEA' ,	1 ,	0 ,	50 ,	'Hotel Artemis') ,
(2 ,	'POOL' ,	1 ,	1 ,	60 ,	'Hotel Artemis'),
(3 ,	'SEA' ,	0 ,	0 ,	70 ,	'Hotel Artemis'),
(4 ,	'SEA' ,	0 ,	0 ,	80 ,	'Hotel Artemis'),
(5 ,	'POOL' ,	0 ,	1 ,	95 ,	'Hotel Artemis'),
(1 ,	'POOL' ,	1 ,	0 ,	40 ,	'Filoxenia') ,
(2 ,	'POOL' ,	1 ,	1 ,	50 ,	'Filoxenia'),
(3 ,	'POOL' ,	0 ,	1 ,	60 ,	'Filoxenia'),
(4 ,	'POOL' ,	1 ,	1 ,	70 ,	'Filoxenia'),
(5 ,	'POOL' ,	0 ,	0 ,	80 ,	'Filoxenia'),
(1 ,	'SEA' ,	1 ,	1 ,	80 ,	'Venus'),

(2 ,	'SEA' ,	1 ,	0 ,	90 ,	'Venus'),
(3 ,	'POOL' ,	1 ,	1 ,	300 ,	'Venus') ,
(4 ,	'SEA' ,	0 ,	0 ,	320 ,	'Venus') ,
(5 ,	'POOL' ,	0 ,	1 ,	350 ,	'Venus') ,
(1 ,	'SEA' ,	1 ,	0 ,	30 ,	'Aiolos') ,
(2 ,	'SEA' ,	1 ,	1 ,	30 ,	'Aiolos') ,
(3 ,	'POOL' ,	1 ,	0 ,	30 ,	'Aiolos') ,
(4 ,	'POOL' ,	0 ,	0 ,	100 ,	'Aiolos') ,
(5 ,	'SEA' ,	0 ,	1 ,	120 ,	'Aiolos') ,
(1 ,	'SEA' ,	1 ,	1 ,	60 ,	'Sorokos') ,
(2 ,	'SEA' ,	1 ,	0 ,	70 ,	'Sorokos') ,
(3 ,	'POOL' ,	1 ,	1 ,	80 ,	'Sorokos') ,
(4 ,	'SEA' ,	0 ,	0 ,	90 ,	'Sorokos') ,
(5 ,	'POOL' ,	1 ,	0 ,	100 ,	'Sorokos') ,
(1 ,	'POOL' ,	1 ,	0 ,	80 ,	'Xenofon') ,
(2 ,	'POOL' ,	1 ,	1 ,	90 ,	'Xenofon') ,
(3 ,	'POOL' ,	1 ,	0 ,	500 ,	'Xenofon') ,
(4 ,	'POOL' ,	0 ,	0 ,	120 ,	'Xenofon') ,
(5 ,	'POOL' ,	1 ,	1 ,	150 ,	'Xenofon') ,
(1 ,	'SEA' ,	1 ,	1 ,	100 ,	'Hotel Stelakis') ,
(2 ,	'SEA' ,	1 ,	0 ,	110 ,	'Hotel Stelakis') ,
(3 ,	'SEA' ,	1 ,	0 ,	130 ,	'Hotel Stelakis') ,
(4 ,	'SEA' ,	0 ,	1 ,	140 ,	'Hotel Stelakis') ,
(5 ,	'POOL' ,	1 ,	0 ,	150 ,	'Hotel Stelakis') ,
(1 ,	'POOL' ,	1 ,	0 ,	80 ,	'Hotel Astoria') ,
(2 ,	'SEA' ,	1 ,	1 ,	85 ,	'Hotel Astoria') ,
(3 ,	'SEA' ,	1 ,	0 ,	95 ,	'Hotel Astoria') ,
(4 ,	'POOL' ,	0 ,	1 ,	110 ,	'Hotel Astoria') ,
(5 ,	'POOL' ,	0 ,	0 ,	300 ,	'Hotel Astoria') ,
(1 ,	'POOL' ,	1 ,	0 ,	90 ,	'Hotel El Greco') ,
(2 ,	'SEA' ,	1 ,	0 ,	115 ,	'Hotel El Greco') ,
(3 ,	'SEA' ,	0 ,	1 ,	125 ,	'Hotel El Greco') ,
(4 ,	'POOL' ,	0 ,	1 ,	140 ,	'Hotel El Greco') ,
(5 ,	'SEA' ,	0 ,	0 ,	820 ,	'Hotel El Greco') ,
(1 ,	'POOL' ,	1 ,	0 ,	90 ,	'Hotel Rinis') ,
(2 ,	'SEA' ,	1 ,	1 ,	100 ,	'Hotel Rinis') ,
(3 ,	'POOL' ,	0 ,	0 ,	120 ,	'Hotel Rinis') ,
(4 ,	'SEA' ,	0 ,	1 ,	140 ,	'Hotel Rinis') ,
(5 ,	'POOL' ,	1 ,	1 ,	170 ,	'Hotel Rinis') ,

(1 , 'SEA' , 0 , 1 , 100 ,	'Hotel Aristeidis') ,
(2 , 'POOL' , 1 , 1 , 130 ,	'Hotel Aristeidis') ,
(3 , 'SEA' , 1 , 1 , 140 ,	'Hotel Aristeidis') ,
(4 , 'SEA' , 0 , 1 , 150 ,	'Hotel Aristeidis') ,
(5 , 'POOL' , 1 , 0 , 170 ,	'Hotel Aristeidis') ,
(1 , 'SEA' , 1 , 0 , 80 ,	'Platon') ,
(2 , 'POOL' , 1 , 1 , 345 ,	'Platon') ,
(3 , 'POOL' , 0 , 1 , 90 ,	'Platon') ,
(4 , 'POOL' , 0 , 0 , 100 ,	'Platon') ,
(5 , 'SEA' , 0 , 0 , 320 ,	'Platon') ,
(1 , 'POOL' , 1 , 0 , 300 ,	'Aristotelis') ,
(2 , 'SEA' , 0 , 0 , 350 ,	'Aristotelis') ,
(3 , 'SEA' , 1 , 0 , 170 ,	'Aristotelis') ,
(4 , 'POOL' , 1 , 0 , 180 ,	'Aristotelis') ,
(5 , 'SEA' , 0 , 0 , 200 ,	'Aristotelis') ,
(1 , 'POOL' , 1 , 0 , 60 ,	'Evripidis') ,
(2 , 'POOL' , 1 , 0 , 75 ,	'Evripidis') ,
(3 , 'POOL' , 1 , 1 , 80 ,	'Evripidis') ,
(4 , 'SEA' , 0 , 0 , 60 ,	'Evripidis') ,
(5 , 'SEA' , 0 , 0 , 210 ,	'Evripidis') ,
(1 , 'POOL' , 0 , 0 , 20 ,	'Sofoklis') ,
(2 , 'SEA' , 1 , 0 , 210 ,	'Sofoklis') ,
(3 , 'SEA' , 1 , 1 , 120 ,	'Sofoklis') ,
(4 , 'SEA' , 0 , 1 , 130 ,	'Sofoklis') ,
(5 , 'POOL' , 0 , 1 , 150 ,	'Sofoklis') ,
(1 , 'POOL' , 1 , 0 , 190 ,	'Sokratis') ,
(2 , 'POOL' , 0 , 0 , 190 ,	'Sokratis') ,
(3 , 'SEA' , 1 , 0 , 100 ,	'Sokratis') ,
(4 , 'SEA' , 1 , 1 , 110 ,	'Sokratis') ,
(5 , 'POOL' , 1 , 0 , 160 ,	'Sokratis') ;

```

INSERT INTO `Basic`.`works` (`Start_Date` , `Finish_Date` , `Position` ,
`Employee_IRS_Number` , `Hotel_id` ) VALUES
('2016-03-18' , '2019-12-31' , 'General Manager' , 1056754 , 'Hilton
Athens' ) ,
('2012-04-13' , '2019-12-31' , 'General Manager' , 1568966 , 'Hilton
Athena' ) ,

```

('2009-11-07' , '2019-12-31' , 'General Manager' , 1898681 , 'Hilton Thessaloniki'),
('2006-08-15' , '2019-12-31' , 'General Manager' , 1752967 , 'Hilton Patra'),
('2013-06-28' , '2019-12-31' , 'General Manager' , 1589876 , 'Hilton Mykonos'),
('2007-01-25' , '2019-12-31' , 'General Manager' , 1976871 , 'Hotel Zeus'),
('2012-05-12' , '2019-12-31' , 'General Manager' , 1358797 , 'Hotel Alexandra'),
('2003-09-22' , '2019-12-31' , 'General Manager' , 1878896 , 'Hotel Ermis'),
('2015-02-09' , '2019-12-31' , 'General Manager' , 1987535 , 'Hotel Apollon'),
('2008-07-30' , '2019-12-31' , 'General Manager' , 1789658 , 'Hotel Artemis'),
('2007-06-16' , '2019-12-31' , 'General Manager' , 1135879 , 'Filoxenia'),
('2017-06-03' , '2019-12-31' , 'General Manager' , 1087687 , 'Venus'),
('2005-03-27' , '2019-12-31' , 'General Manager' , 1354478 , 'Aiolos'),
('2001-02-21' , '2019-12-31' , 'General Manager' , 1489576 , 'Sorokos'),
('2013-05-23' , '2019-12-31' , 'General Manager' , 1791632 , 'Xenofon'),
('2008-07-29' , '2019-12-31' , 'General Manager' , 1887523 , 'Hotel Stelakis'),
('2006-05-27' , '2019-12-31' , 'General Manager' , 1875634 , 'Hotel Astoria'),
('2003-02-11' , '2019-12-31' , 'General Manager' , 1985315 , 'Hotel El Greco'),
('2004-03-11' , '2019-12-31' , 'General Manager' , 1246879 , 'Hotel Rinis'),
('2009-06-27' , '2019-12-31' , 'General Manager' , 1576575 , 'Hotel Aris'),
('2015-07-12' , '2019-12-31' , 'General Manager' , 1357854 , 'Platon'),
('2017-03-15' , '2019-12-31' , 'General Manager' , 1862368 , 'Aristotelis'),
('2009-06-26' , '2019-12-31' , 'General Manager' , 1451254 , 'Sofoklis'),
('2008-10-10' , '2019-12-31' , 'General Manager' , 1852543 , 'Evripidis'),
('2004-09-01' , '2019-12-31' , 'General Manager' , 1399845 , 'Sokratis'),
('2009-06-14' , '2019-12-31' , 'Cleaner' , 1542956 , 'Sorokos'),
('2003-08-05' , '2019-12-31' , 'Gardener' , 1796547 , 'Venus'),
('2007-06-06' , '2019-12-31' , 'Laundry Manager' , 1487455 , 'Sofoklis'),
('2004-09-22' , '2019-12-31' , 'Receptionist' , 1834120 , 'Filoxenia'),
('2006-10-11' , '2019-12-31' , 'Receptionist' , 1796541 , 'Hilton Mykonos');

```
INSERT INTO `Basic`.`Hotel_Email_Address` (`Email` , `Hotel_id` )  
VALUES
```

```
( 'hiltonathens@gmail.com' ,      'Hilton Athens' ),  
( 'hiltonathena@gmail.com' ,      'Hilton Athena' ),  
( 'hiltonthess@gmail.com' ,      'Hilton Thessaloniki' ),  
( 'hiltonpatra@gmail.com' ,      'Hilton Patra' ),  
( 'hiltonmykonos@gmail.com' ,      'Hilton Mykonos' ),  
( 'hotelzeus@gmail.com' ,      'Hotel Zeus' ),  
( 'hotelalexandra@gmail.com' , 'Hotel Alexandra' ),  
( 'hotelermis@gmail.com' ,      'Hotel Ermis' ),  
( 'hotelapollon@gmail.com' ,      'Hotel Apollon' ),  
( 'hotelartemis@gmail.com' ,      'Hotel Artemis' ),  
( 'hotelfiloxenia@gmail.com' ,      'Filoxenia' ),  
( 'hotelvenus@gmail.com' ,      'Venus' ),  
( 'hotelaiolos@gmail.com' ,      'Aiolos' ),  
( 'hotelsorokos@gmail.com' ,      'Sorokos' ),  
( 'hotelxenofon@gmail.com' ,      'Xenofon' ),  
( 'hotelstelakis@gmail.com' ,      'Hotel Stelakis' ),  
( 'hotelastoria@gmail.com' ,      'Hotel Astoria' ),  
( 'hotelelgreco@gmail.com' ,      'Hotel El Greco' ),  
( 'hotelrinis@gmail.com' ,      'Hotel Rinis' ),  
( 'hotelaris@gmail.com' ,      'Hotel Aris' ),  
( 'hotelplaton@gmail.com' ,      'Platon' ),  
( 'hotelaristotelis@gmail.com' , 'Aristotelis' ),  
( 'hotelsofoklis@gmail.com' ,      'Sofoklis' ),  
( 'hotelevripidis@gmail.com' ,      'Evripidis' ),  
( 'hotelsokratis@gmail.com' ,      'Sokratis' );
```

```
INSERT INTO `Basic`.`Hotel_Phone_Number` (`Phone` , `Hotel_id` )  
VALUES
```

```
(2102838457, 'Hilton Athens' ),  
(2102986074, 'Hilton Athena' ),  
(2310963854, 'Hilton Thessaloniki' ),  
(2520686543, 'Hilton Patra' ),  
(2315985321, 'Hilton Mykonos' ),  
(2895623145, 'Hotel Zeus' ),
```

```

(2852389856, 'Hotel Alexandra' ),
(2345854652, 'Hotel Ermis' ),
(2956459896, 'Hotel Apollon' ),
(2186584689, 'Hotel Artemis' ),
(2105578596, 'Filoxenia' ),
(2120547845, 'Venus' ),
(2165421544, 'Aiolos' ),
(2850532546, 'Sorokos' ),
(2102899874, 'Xenofon' ),
(2874521454, 'Hotel Stelakis' ),
(2785412549, 'Hotel Astoria' ),
(2810598564, 'Hotel El Greco' ),
(2845532486, 'Hotel Rinis' ),
(2164512545, 'Hotel Aris' ),
(2186455643, 'Platon' ),
(2874525467, 'Aristotelis' ),
(2187541542, 'Sofoklis' ),
(2867523124, 'Evripidis' ),
(2154345267, 'Sokratis' );

```

```

INSERT INTO `Basic`.`Reserves` (`Start_Date`, `Finish_Date`,
`Customer_IRS_Number`, `Room_id`, `Hotel_id`) VALUES
('2018-06-02', '2018-06-08', 1324657, 6, 'Hilton Athena');

```

```

INSERT INTO `Basic`.`Hotel_Group_Email_Address` (`Email` ,
`Hotel_Group_id` ) VALUES
( 'hilton@gmail.com' , 'Hilton' ),
( 'sanihotels@gmail.com' , 'Sani Hotels' ),
( 'grestbritain@gmail.com' , 'Great Britain' ),
( 'hotelpalaces@gmail.com' , 'Hotel Palaces' ),
( 'resortsparadise@gmail.com' , 'Resorts Paradise' );

```

```

INSERT INTO `Basic`.`Hotel_Group_Phone_Number` ( `Phone` ,
`Hotel_Group_id` ) VALUES
( 2108986552 , 'Hilton' ),
( 2310896532 , 'Sani Hotels' ),
( 2895652345 , 'Great Britain' ),
( 2165565956 , 'Hotel Palaces' ),

```

(2899865235 , 'Resorts Paradise');

```
INSERT INTO `Basic`.`Amenities` ( `Room_id`, `Amenity` ) VALUES
( 1 , 'wifi' ),
( 1 , 'tv' ),
( 2 , 'tv' ),
( 3 , 'tv' ),
( 3 , 'wifi' ),
( 4 , 'tv' ),
(
  4 , 'wifi' ),
( 5 , 'tv' ),
( 6 , 'tv' ),
( 7 , 'tv' ),
( 7 , 'wifi' ),
( 8 , 'wifi' ),
( 8 , 'tv' ),
( 9 , 'tv' ),
( 10 , 'wifi' ),
( 11 , 'wifi' ),
( 12 , 'tv' ),
( 13 , 'tv' ),
( 13 , 'wifi' ),
( 14 , 'wifi' ),
( 14 , 'tv' ),
( 15 , 'wifi' ),
( 15 , 'tv' ),
( 16 , 'tv' ),
( 17 , 'wifi' ),
( 18 , 'wifi' ),
( 19 , 'tv' ),
( 20 , 'wifi' ),
( 21 , 'wifi' ),
( 21 , 'tv' ),
( 22 , 'wifi' ),
( 22 , 'tv' ),
( 23 , 'tv' ),
( 24 , 'wifi' ),
```

(24 , 'tv'),
(25 , 'wifi'),
(26 , 'wifi'),
(26 , 'tv'),
(27 , 'wifi'),
(28 , 'wifi'),
(29 , 'tv'),
(30 , 'tv'),
(30 , 'wifi'),
(31 , 'tv'),
(32 , 'wifi'),
(33 , 'wifi'),
(34 , 'wifi'),
(34 , 'tv'),
(35 , 'tv'),
(35 , 'wifi'),
(36 , 'wifi'),
(36 , 'tv'),
(37 , 'wifi'),
(37 , 'tv'),
(38 , 'wifi'),
(39 , 'tv'),
(40 , 'tv'),
(41 , 'wifi'),
(41 , 'tv'),
(42 , 'tv'),
(43 , 'wifi'),
(44 , 'wifi'),
(44 , 'tv'),
(45 , 'wifi'),
(45 , 'tv'),
(46 , 'wifi'),
(46 , 'tv'),
(47 , 'wifi'),
(47 , 'tv'),
(48 , 'wifi'),
(49 , 'tv'),
(50 , 'tv'),
(51 , 'tv'),

(51 , 'wifi'),
(52 , 'wifi'),
(53 , 'tv'),
(53 , 'wifi'),
(54 , 'wifi'),
(54 , 'tv'),
(55 , 'tv'),
(55 , 'wifi'),
(56 , 'tv'),
(57 , 'wifi'),
(58 , 'wifi'),
(59 , 'tv'),
(60 , 'wifi'),
(60 , 'tv'),
(61 , 'tv'),
(62 , 'tv'),
(63 , 'wifi'),
(64 , 'tv'),
(65 , 'tv'),
(66 , 'wifi'),
(66 , 'tv'),
(67 , 'tv'),
(68 , 'wifi'),
(68 , 'tv'),
(69 , 'wifi'),
(70 , 'wifi'),
(70 , 'tv'),
(71 , 'tv'),
(71 , 'wifi'),
(72 , 'wifi'),
(72 , 'tv'),
(73 , 'tv'),
(74 , 'wifi'),
(75 , 'tv'),
(75 , 'wifi'),
(76 , 'tv'),
(77 , 'wifi'),
(78 , 'tv'),
(79 , 'wifi'),

(80 , 'wifi'),
(80 , 'tv'),
(81 , 'wifi'),
(81 , 'tv'),
(82 , 'tv'),
(82 , 'wifi'),
(83 , 'tv'),
(84 , 'wifi'),
(85 , 'wifi'),
(86 , 'tv'),
(87 , 'wifi'),
(87 , 'tv'),
(88 , 'tv'),
(88 , 'wifi'),
(89 , 'tv'),
(90 , 'wifi'),
(90 , 'tv'),
(91 , 'wifi'),
(92 , 'tv'),
(92 , 'wifi'),
(93 , 'tv'),
(93 , 'wifi'),
(94 , 'tv'),
(94 , 'wifi'),
(95 , 'wifi'),
(95 , 'tv'),
(96 , 'wifi'),
(96 , 'tv'),
(97 , 'wifi'),
(98 , 'tv'),
(99 , 'wifi'),
(100 , 'wifi'),
(101 , 'tv'),
(102 , 'tv'),
(103 , 'wifi'),
(103 , 'tv'),
(104 , 'wifi'),
(104 , 'tv'),
(105 , 'wifi'),


```
( 106,      'tv'   ),  
( 106, 'wifi'   ),  
( 107,      'tv'   ),  
( 107,      'wifi'   ),  
( 108,      'tv'   ),  
( 109,      'wifi'   ),  
( 109,      'tv'   ),  
( 110,      'wifi'   ),  
( 111,      'tv'   ),  
( 111,      'wifi'   ),  
( 112,      'tv'   ),  
( 113,      'wifi'   ),  
( 114,      'tv'   ),  
( 114,      'wifi'   ),  
( 115,      'tv'   ),  
( 115,      'wifi'   ),  
( 116,      'tv'   ),  
( 117,      'tv'   ),  
( 118,      'wifi'   ),  
( 119,      'wifi' ),  
( 120,      'tv'   ),  
( 120,      'wifi'   ),  
( 121,      'tv'   ),  
( 122,      'wifi'   ),  
( 122,      'tv'   ),  
( 123,      'wifi'   ),  
( 124,      'tv'   ),  
( 124,      'wifi' ),  
( 125,      'tv'   ),  
( 125,      'wifi'   );
```

```
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

4. Όλα τα ζητούμενα αρχεία κώδικα περιέχονται στο αρχείο Code.zip που επισυνάπτεται.

