

ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

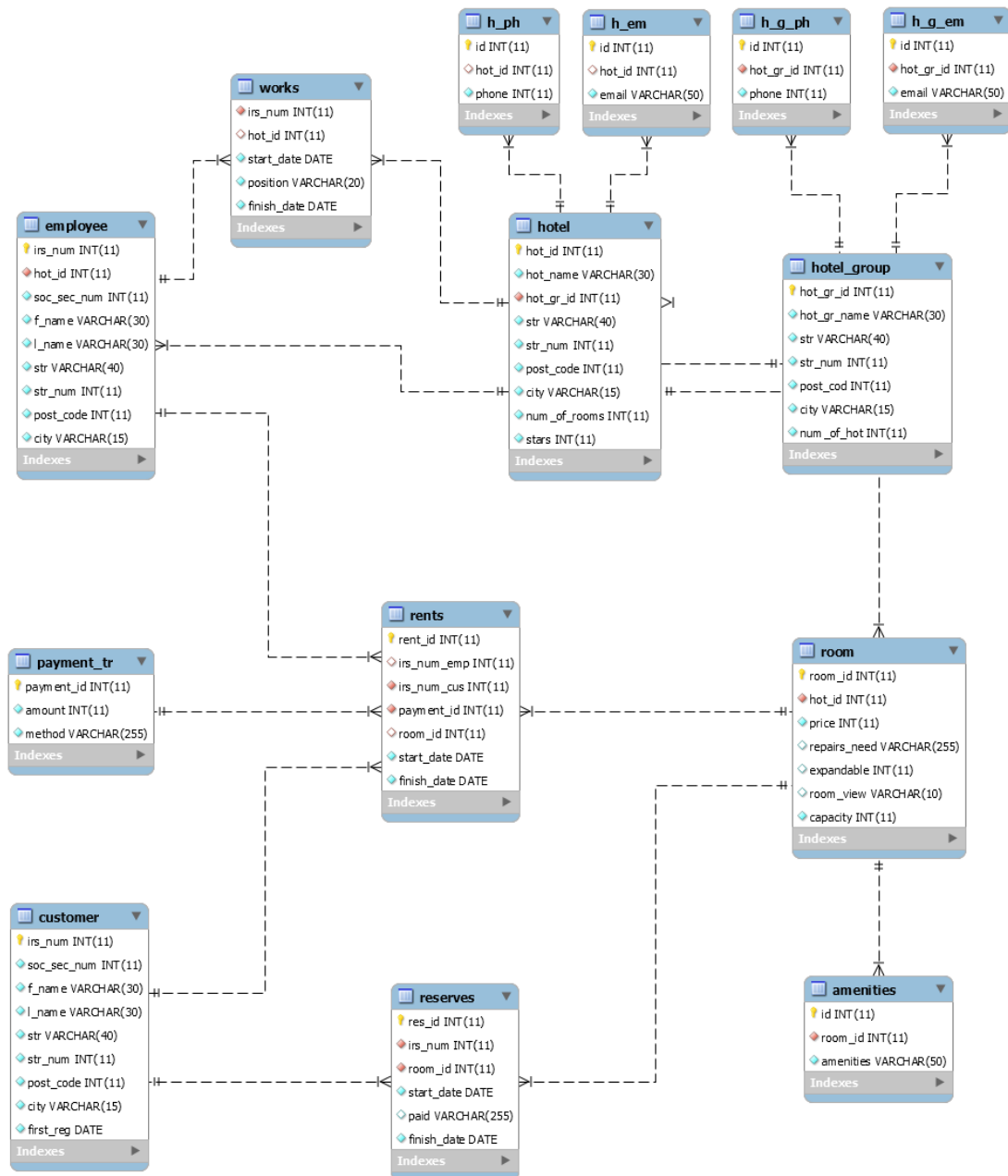
ΕΞΑΜΗΝΙΑΙΑ ΕΡΓΑΣΙΑ project

Πολύβιος Παπακωνσταντίνου

A.M. 03114892

22/5/2018

Το διάγραμμα της βάσης:



Ο κώδικας:

```
CREATE TABLE HOTEL_GROUP (  
    hot_gr_id INT NOT NULL AUTO_INCREMENT,  
    hot_gr_name varchar(30) NOT NULL,  
    str varchar(40) NOT NULL,  
    str_num INT NOT NULL,  
    post_cod INT NOT NULL,  
    city varchar(15) NOT NULL,  
    num_of_hot INT NOT NULL,  
    UNIQUE(hot_gr_name),  
    PRIMARY KEY(hot_gr_id)  
);  
  
CREATE TABLE H_G_PH (  
    id INT NOT NULL AUTO_INCREMENT,  
    hot_gr_id INT NOT NULL,  
    phone INT NOT NULL,  
    PRIMARY KEY(id),  
  
    FOREIGN KEY(hot_gr_id) REFERENCES HOTEL_GROUP(hot_gr_id) ON  
UPDATE CASCADE ON DELETE CASCADE  
);  
  
CREATE TABLE H_G_EM (  
    id INT NOT NULL AUTO_INCREMENT,  
    hot_gr_id INT NOT NULL,  
    email varchar(50) NOT NULL,  
    PRIMARY KEY(id),  
  
    FOREIGN KEY(hot_gr_id) REFERENCES HOTEL_GROUP(hot_gr_id) ON  
UPDATE CASCADE ON DELETE CASCADE  
);  
  
CREATE TABLE HOTEL (  
    hot_id INT NOT NULL AUTO_INCREMENT,  
    hot_name varchar(30) NOT NULL,  
    hot_gr_id INT NOT NULL,  
    str varchar(40) NOT NULL,  
    str_num INT NOT NULL,  
    post_code INT NOT NULL,  
    city varchar(15) NOT NULL,  
    num_of_rooms INT NOT NULL,  
    stars INT NOT NULL,  
    PRIMARY KEY(hot_id),  
    UNIQUE(hot_name),  
  
    FOREIGN KEY(hot_gr_id) REFERENCES HOTEL_GROUP(hot_gr_id) ON  
UPDATE CASCADE ON DELETE CASCADE  
);  
  
CREATE TABLE H_PH (  
    id INT NOT NULL AUTO_INCREMENT,  
    hot_id INT,  
    phone INT NOT NULL,  
    PRIMARY KEY(id),  
  
    FOREIGN KEY(hot_id) REFERENCES HOTEL(hot_id) ON UPDATE CASCADE ON  
DELETE CASCADE  
);
```

```

CREATE TABLE H_EM (
    id INT NOT NULL AUTO_INCREMENT,
    hot_id INT,
    email varchar(50) NOT NULL,
    PRIMARY KEY(id),

    FOREIGN KEY(hot_id) REFERENCES HOTEL(hot_id) ON UPDATE CASCADE ON
DELETE CASCADE
);

CREATE TABLE ROOM (
    room_id INT NOT NULL,
    hot_id INT NOT NULL,
    price INT NOT NULL,
    repairs_need varchar(255),
    expandable INT,
    room_view varchar(10),
    capacity INT NOT NULL,
    PRIMARY KEY(room_id),

    FOREIGN KEY(hot_id) REFERENCES HOTEL(hot_id) ON UPDATE CASCADE ON
DELETE CASCADE
);

CREATE TABLE AMENITIES (
    id INT NOT NULL AUTO_INCREMENT,
    room_id INT NOT NULL,
    amenities varchar(50) NOT NULL,
    PRIMARY KEY(id),

    FOREIGN KEY(room_id) REFERENCES ROOM(room_id) ON UPDATE CASCADE
ON DELETE CASCADE
);

CREATE TABLE EMPLOYEE (
    irs_num INT NOT NULL,
    hot_id INT NOT NULL,
    soc_sec_num INT NOT NULL,
    f_name varchar(30) NOT NULL,
    l_name varchar(30) NOT NULL,
    str varchar(40) NOT NULL,
    str_num INT NOT NULL,
    post_code INT NOT NULL,
    city varchar(15) NOT NULL,
    PRIMARY KEY(irs_num),

    FOREIGN KEY(hot_id) REFERENCES HOTEL(hot_id) ON UPDATE CASCADE ON
DELETE CASCADE
);

CREATE TABLE CUSTOMER (
    irs_num INT NOT NULL,
    soc_sec_num INT NOT NULL,
    f_name varchar(30) NOT NULL,
    l_name varchar(30) NOT NULL,
    str varchar(40) NOT NULL,
    str_num INT NOT NULL,
    post_code INT NOT NULL,
    city varchar(15) NOT NULL,
    first_reg DATE NOT NULL,
    PRIMARY KEY(irs_num)

```

);

```
CREATE TABLE WORKS (  
    irs_num INT NOT NULL,  
    hot_id INT,  
    start_date DATE NOT NULL,  
    position varchar(20) NOT NULL,  
    finish_date DATE NOT NULL,  
    UNIQUE(irs_num, hot_id),  
  
    CONSTRAINT CHK_dates CHECK (finish_date > start_date),  
    FOREIGN KEY(irs_num) REFERENCES EMPLOYEE(irs_num) ON UPDATE  
CASCADE ON DELETE CASCADE,  
    FOREIGN KEY(hot_id) REFERENCES HOTEL(hot_id) ON UPDATE CASCADE ON  
DELETE CASCADE  
);
```

```
CREATE TABLE RESERVES (  
    res_id INT NOT NULL AUTO_INCREMENT,  
    irs_num INT NOT NULL,  
    room_id INT NOT NULL,  
    start_date DATE NOT NULL,  
    paid varchar(255),  
    finish_date DATE NOT NULL,  
  
    PRIMARY KEY(res_id),  
    CONSTRAINT CHK_dates CHECK (finish_date > start_date),  
    FOREIGN KEY(irs_num) REFERENCES CUSTOMER(irs_num) ON UPDATE  
CASCADE ON DELETE CASCADE,  
    FOREIGN KEY(room_id) REFERENCES ROOM(room_id) ON UPDATE CASCADE  
ON DELETE CASCADE  
);
```

```
CREATE TABLE PAYMENT_TR (  
    payment_id INT NOT NULL AUTO_INCREMENT,  
    amount INT NOT NULL,  
    method varchar(255) NOT NULL,  
  
    PRIMARY KEY(payment_id)  
);
```

```
CREATE TABLE RENTS (  
    rent_id INT NOT NULL AUTO_INCREMENT,  
    irs_num_emp INT,  
    irs_num_cus INT NOT NULL,  
    payment_id INT NOT NULL,  
    room_id INT,  
    start_date DATE NOT NULL,  
    finish_date DATE NOT NULL,  
  
    PRIMARY KEY(rent_id),  
    CONSTRAINT CHK_dates CHECK (finish_date > start_date),  
    FOREIGN KEY(irs_num_emp) REFERENCES EMPLOYEE(irs_num) ON UPDATE  
CASCADE ON DELETE SET NULL,  
    FOREIGN KEY(irs_num_cus) REFERENCES CUSTOMER(irs_num) ON UPDATE  
CASCADE ON DELETE RESTRICT,  
    FOREIGN KEY(payment_id) REFERENCES PAYMENT_TR(payment_id) ON  
UPDATE CASCADE ON DELETE RESTRICT,  
    FOREIGN KEY(room_id) REFERENCES ROOM(room_id) ON UPDATE CASCADE  
ON DELETE SET NULL  
);
```

Οι περιορισμοί που έχουν οριστεί χωρίζονται στις εξής κατηγορίες:

- **NOT NULL**

Επιλέχθηκε για όσα δε θέλαμε να μένουν κενά, είτε επειδή θα επηρέαζαν τη λειτουργικότητα της ιστοσελίδας (f_name/ l_name για customers/ employees), είτε επειδή δε θα ήταν «ρεαλιστικό» για τη φύση της βάσης (method για payment_tr).

- **UNIQUE**

Κυρίως επιλέχθηκε για την λειτουργικότητα της ιστοσελίδας και την αποφυγή σύγχυσης του χρήστη. Για παράδειγμα, θέλουμε τα ονόματα των ξενοδοχείων να είναι μοναδικά, προκειμένου ο χρήστης να μπορεί να επιλέξει όποιο θέλει χωρίς να μπερδευτεί.

- **CHECK _DATES**

Σε ορισμένους πίνακες υπάρχουν «ζευγάρια» ημερομηνιών, όπως άφιξης – αναχώρησης. Το συγκεκριμένο constraint ορίστηκε για να διασφαλιστεί πως οι ημερομηνίες αυτές μπήκαν με τη σωστή σειρά στη βάση.

Σχετικά με τις ημερομηνίες και την ορθότητα τους, γίνεται έλεγχος και μέσα στην ιστοσελίδα, για παράδειγμα όταν ο χρήστης κάνει αναζήτηση δωματίου, ελέγχεται αν η ημερομηνία άφιξης έχει περάσει.

- **ON DELETE/UPDATE CASCADE/RESTRICT/SET NULL**

Όσον αφορά τα updates, σε όλα επιλέχθηκε το cascade, προκειμένου ο πίνακας παιδί να ενημερώνεται για αλλαγές στον πίνακα πατέρα του.

Για τα deletes, επιλέχθηκε το cascade όταν θέλουμε η διαγραφή του πίνακα πατέρα να οδηγεί στη διαγραφή του πίνακα παιδιού (όταν διαγράψουμε ένα ξενοδοχείο από τη βάση δεν υπάρχει λόγος να κρατάμε δεδομένα για τα δωμάτια του).

Το set null, επιλέχθηκε στην περίπτωση που η διαγραφή του πίνακα πατέρα δεν «αφορά» τον πίνακα παιδί. Εάν διαγραφεί πχ ένας εργαζόμενος, δε μας ενδιαφέρει για τις κρατήσεις που είχε αναλάβει, άρα στο αντίστοιχο πεδίο της RENTS, θέτουμε NULL.

Το restrict επιλέχθηκε όταν θέλαμε να διασφαλίσουμε πως ένα record του πίνακα πατέρα δε μπορεί να διαγραφεί, εφόσον σχετίζεται με record στον πίνακα παιδί. Για παράδειγμα, δε θέλουμε ποτέ να διαγραφεί πελάτης από τον πίνακα CUSTOMER, εφόσον έχει γίνει κράτηση στο όνομα του.

Ορίστηκαν 3 ευρετήρια, για τους πίνακες HOTEL, ROOMS και RESERVES. Για τους πρώτους 2 επειδή ο πελάτης βλέπει όλα τα ξενοδοχεία της βάσης στην καρτέλα «Πληροφορίες ξενοδοχείων» και επειδή αναζητεί για δωμάτια.

Για τον reserves δημιουργήθηκε ευρετήριο επειδή όταν συνδέεται υπάλληλος, στην αρχική σελίδα, υπάρχει πίνακας με τις εκκρεμείς κρατήσεις που έχουν γίνει.

Επίσης ορίστηκε ένα trigger, για τη διαγραφή στον πίνακα EMPLOYEES, εφόσον έχει προηγηθεί διαγραφή από τον πίνακα WORKS.

Το site είναι «χωρισμένο» σε 2 μέρη. Για πελάτες, οι οποίοι μπορούν να δουν κάποιες πληροφορίες για τα ξενοδοχεία και για τους υπαλλήλους, οι οποίοι μπορούν και να επεξεργαστούν (inserts/updates/deletes) στοιχεία της βάσης.

Όσον αφορά την αναζήτηση δωματίου από πελάτη, καλείται να συμπληρώσει τα εξής πεδία: ΑΦΜ, ημερομηνίες άφιξης/αναχώρησης, αριθμός ατόμων, πόλη, ξενοδοχείο, κατηγορία και παροχές.

Από τα παραπάνω πεδία, υποχρεωτικά είναι μόνο αυτά των ημερομηνιών. Προτιμήθηκε η επιλογή για ξενοδοχείο αντί αλυσίδας για να είναι πιο ρεαλιστική η αναζήτηση.

Στις παροχές, ο πελάτης μπορεί να συμπληρώσει πως θέλει δωμάτιο με μπαλκόνι (mpalkoni), πρωινό (prwino), κουζίνα (kouzina), spa ή πισίνα (pisina).

Έπειτα, εφόσον υπάρχουν διαθέσιμα δωμάτια που πληρούν τα κριτήρια που έθεσε, του εμφανίζεται λίστα με αυτά και αρκεί να πατήσει πάνω στην τιμή για να κάνει την κράτηση που θέλει.

Να σημειωθεί πως αν δεν έχει εισάγει το ΑΦΜ του στην προηγούμενη φόρμα θα του εμφανίσει κατάλληλο μήνυμα.

Για το project χρησιμοποιήθηκαν mysql server 5.7.2, php 7.2 και apache 2.4.

Προκειμένου να εγκαταστήσει κάποιος την εφαρμογή απ την αρχή, θα χρειαστεί να «τρέξει» τα αρχεία hotelia.sql και data.sql (για το filling της βάσης με εικονικά στοιχεία).

Το hotelia.sql δημιουργεί εκ νέου τη βάση spring_project.

Σημειώνεται πως κάθε κομμάτι κώδικα php του site συνδέεται στη βάση με το username: Don και password: password.