

ΤΕΧΝΟΛΟΓΙΑ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

ΕΦΑΡΜΟΓΗ ΞΕΝΟΔΟΧΕΙΑΚΩΝ ΚΡΑΤΗΣΕΩΝ

Επιμέλεια: Γάλλε Μιχαηλία 185158, Λεοπάρντι Μάρκος 154484

ΠΕΡΙΕΧΟΜΕΝΑ

1. ΕΙΣΑΓΩΓΗ	1
2. ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ	2
3. FUNCTIONS SQL	4
3.1. ΕΙΣΑΓΩΓΗ	4
3.2. ΑΝΑΖΗΤΗΣΗ	7
3.3. ΕΠΙΚΑΙΡΟΠΟΙΗΣΗ	9
3.4. ΔΙΑΓΡΑΦΗ	10
4. ΚΩΔΙΚΑΣ ΔΙΕΠΑΦΗΣ	11
5. ΔΙΕΠΑΦΗ	13

1. ΕΙΣΑΓΩΓΗ

Στα πλαίσια του μαθήματος Τεχνολογία Βάσεων Δεδομένων, δημιουργήσαμε μία εφαρμογή ξενοδοχειακών κρατήσεων, από την μεριά της επιχείρησης. Η εφαρμογή δημιουργήθηκε με σκοπό την κάλυψη της απαίτησης για ένα πληροφοριακό σύστημα μηχανογράφησης ξενοδοχείου. Η βάση δεδομένων υλοποιήθηκε με την χρήση του PgAdmin III, σε γλώσσα PostgreSQL. Το GUI υλοποιήθηκε στο NetBeans, στη γλώσσα Java Swing. Η εφαρμογή στοχεύει στην διευκόλυνση των ξενοδοχοϋπαλλήλων να δημιουργούν, να επεξεργάζονται, να ολοκληρώνουν και να διαγράφουν κρατήσεις, με όσο λιγότερα βήματα είναι δυνατό, και με αυτοματοποιημένες διαδικασίες επιλογής δωματίου, καθώς και να κρατούν ιστορικό των κρατήσεων και των πελατών.

2. ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Η βάση δεδομένων αποτελείται από 6 πίνακες δεδομένων. Οι πίνακες είναι οι εξής:

- Guests αναφέρεται στους πελάτες του ξενοδοχείου.
- Rooms αναφέρεται στο σύνολο των δωματίων που υπάρχουν στο ξενοδοχείο.
- Room_Categories αναφέρεται στα είδη των δωματίων (μονόκλινα, δίκλινα, τρίκλινα).
- Reserved_Rooms αναφέρεται στις κρατήσεις που έχουν γίνει και είναι ενεργές.
- Past_reservations αναφέρεται στο ιστορικό κρατήσεων που έχουν ολοκληρωθεί.
- Log_table αναφέρεται στο ιστορικό των ενεργειών που έχουν γίνει στη βάση.

Οι πίνακες Room_Categories και Rooms έχουν σταθερά rows και δεν αλλάζουν. Ο πίνακας Room_Categories αποτελείται από τα columns categoryid τύπου int και categoryname τύπου varchar. Ο πίνακας Rooms αποτελείται από τα columns idroom τύπου int, το οποίο δηλώνει και τον αριθμό δωματίου και είναι το primary key, από το roomcategory τύπου int, το οποίο είναι foreign key του room_categories και αντιστοιχεί στο column categoryid, και από το price τύπου numeric, που δηλώνει την τιμή του δωματίου.

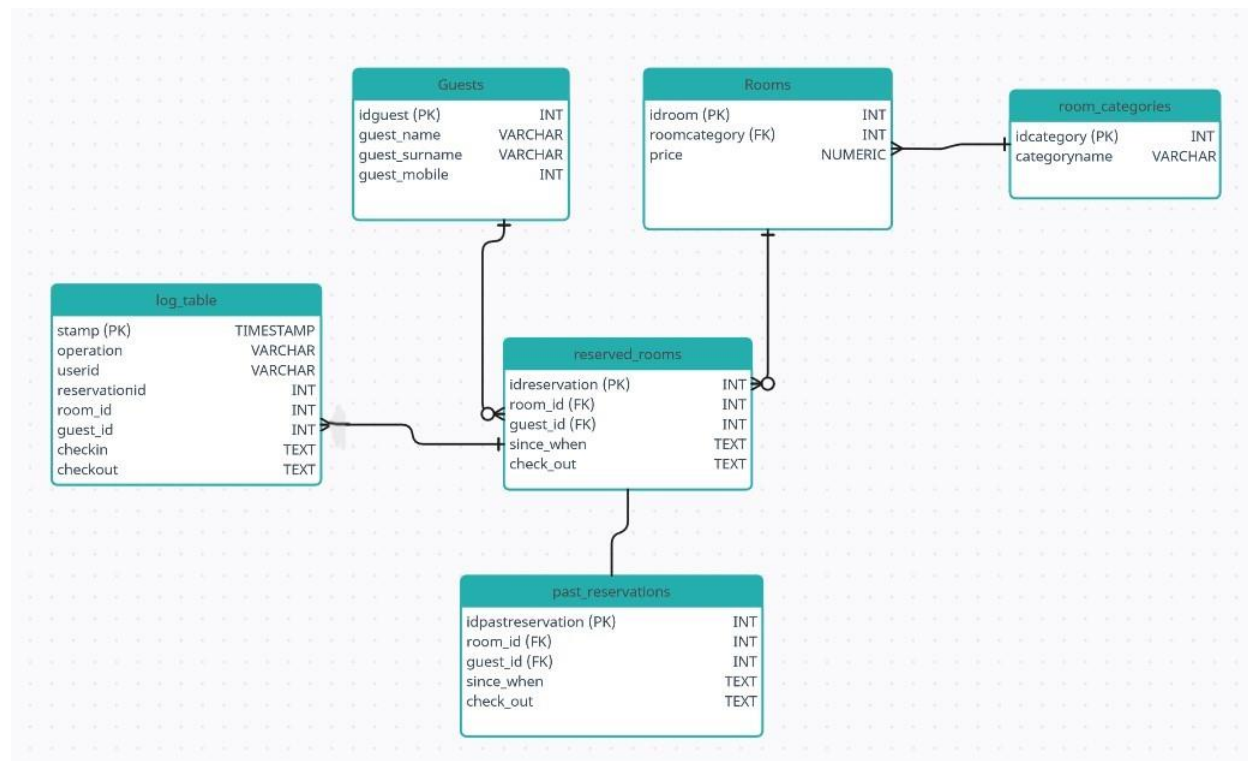
Ο πίνακας guest αποτελείται από τα columns idguest τύπου int, το οποίο αναφέρει τον αριθμό ταυτότητας του πελάτη, το guest_name, guest_surname και guest_mobile, τα οποία δηλώνουν το όνομα, επίθετο και κινητό του πελάτη αντίστοιχα. Ο πίνακας guests, συμπληρώνεται και επικαιροποιείται αυτόματα, με τα κατάλληλα procedures, μετά από κάθε δημιουργία ή επεξεργασία κάποιας κράτησης στον πίνακα reserved_rooms.

Ο πίνακας reserved_rooms αποτελείται από τα columns idreservation τύπου int, δηλώνει τον μοναδικό αριθμό της κάθε κράτησης, είναι το primary key και κάνει auto increment. Τα columns room_id και guest_id είναι foreign keys και αναφέρονται στα columns idroom του πίνακα rooms και idguest του πίνακα guests αντίστοιχα. Και τα columns since_when και check_out τύπου text αναφέρονται στις ημερομηνίες της κράτησης. Όταν γίνεται διαγραφή μιας κράτησης, αντιστοιχείται στην ακύρωση της, και δεν ακολουθεί καμία άλλη ενέργεια. Στην περίπτωση της ολοκλήρωσης μιας κράτησης, δηλαδή ότι πραγματοποιήθηκε και ο πελάτης έχει φύγει πλέον, η κράτηση αντιγράφεται στον πίνακα past_reservations σαν ιστορικό, και διαγράφεται από τον πίνακα reserved_rooms ο οποίος αποτελείται μόνο από ενεργές κρατήσεις. Ο πίνακας past_reservations είναι ένα αντίγραφο του πίνακα reserved_rooms και λειτουργεί ως πίνακας ιστορικού παλαιών κρατήσεων.

Ο πίνακας log_table είναι ο πίνακας που εξυπηρετεί την ανάγκη για log file. Αποτελείται από τα columns stamp τύπου timestamp που δέχεται την ώρα εισαγωγής του row και είναι το primary key, operation τύπου varchar που δέχεται ένα γράμμα I, U, D για insert, update και delete

αντίστοιχα, ανάλογα με την ενέργεια που έκανε trigger την εισαγωγή της νέας σειράς. Το column userid τύπου varchar που δηλώνει τον χρήστη που πραγματοποίησε την ενέργεια στη βάση, και τα columns room_id, guest_id, checkin, checkout που παίρνουν τις παλιές ή νέες τιμές του πίνακα reserved_rooms ανάλογα με την ενέργεια που πραγματοποιήθηκε.

Όπως φαίνεται και στο διάγραμμα (εικόνα 1) η σχέση Guest και reserved_rooms είναι one to zero or many, αφού ένας πελάτης μπορεί να συσχετίζεται με ένα, κανένα ή παραπάνω rows του πίνακα reserved_rooms. Την ίδια σχέση έχουν και ο πίνακας rooms reserved_rooms. Ο πίνακας room_categories και rooms έχουν σχέση one to many, αφού μία κατηγορία του πίνακα room_categories μπορεί να αντιστοιχεί σε πολλά δωμάτια του πίνακα rooms. Οι πίνακες reserved_rooms και log_table έχουν σχέση one to many, αφού κάθε row του πίνακα reserved_rooms συσχετίζεται με ένα ή παραπάνω rows του πίνακα log_table. Ο πίνακας past_reservations είναι αντίγραφο του πίνακα reserved_rooms, ωστόσο καμία σειρά των πινάκων δεν συσχετίζεται με κάποια άλλη, αφού το past_reservations λειτουργεί σαν παλιά εικόνα του reserved_rooms.



Εικόνα 1: Διάγραμμα ροής βάσης δεδομένων

3. FUNCTIONS SQL

Για την διατήρηση της ασφάλειας της βάσης δεδομένων μέσω της διεπαφής, δεν αναφέρονται ονόματα πινάκων ή στηλών στον κώδικα. Υπάρχουν κατάλληλες διαδικασίες, που ονομάζονται functions, αποθηκευμένες στη βάση, οι οποίες καλούνται στον κώδικα για να εκτελεστούν οι αντίστοιχες ενέργειες.

3.1. ΕΙΣΑΓΩΓΗ

Οι παρακάτω διαδικασίες εκτελούν την εισαγωγή δεδομένων σε πίνακες.

```
CREATE OR REPLACE FUNCTION createreservation(  
  g_name character varying,  
  g_surname character varying,  
  g_id integer,  
  g_mobile text,  
  room_type integer,  
  checkin text,  
  checkout text)  
RETURNS void AS  
$BODY$  
DECLARE  
  temp_room INT;  
  
BEGIN  
  -- Den dexetai to ON CONFLICT ayto to version, opote sindiazo update & exception  
  BEGIN  
    INSERT INTO guests (guest_name, guest_surname, idguest, guest_mobile)  
    VALUES (g_name, g_surname, g_id, g_mobile);  
  EXCEPTION WHEN unique_violation THEN  
    UPDATE guests  
    SET  
      guest_name = g_name,  
      guest_surname = g_surname,  
      guest_mobile = g_mobile  
    WHERE idguest = g_id;  
  END;  
  
  SELECT r.idroom INTO temp_room  
  FROM rooms r  
  LEFT JOIN reserved_rooms rr ON r.idroom = rr.room_id  
  WHERE r.roomcategory = room_type  
  AND rr.room_id IS NULL  
  AND NOT EXISTS (  
    SELECT 1  
    FROM reserved_rooms
```

```

WHERE r.idroom = room_id
AND to_timestamp(since_when, 'DD-MM-YYYY') <= to_timestamp(checkout, 'DD-MM-YYYY')
AND to_timestamp(check_out, 'DD-MM-YYYY') >= to_timestamp(checkin, 'DD-MM-YYYY')
)
ORDER BY random()
LIMIT 1;

--elegxos epeidi an den vrei diathesimo domatio epistrefei null kai to idroom den dexetai timi null
IF temp_room IS NOT NULL THEN

    INSERT INTO reserved_rooms (room_id, guest_id, since_when, check_out)
    VALUES (temp_room, g_id, checkin, checkout);
END IF;

IF temp_room IS NULL THEN
    RAISE EXCEPTION 'No available room for reservation.';
END IF;
END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION createreservation(character varying, character varying, integer, text, integer, text,
text)
OWNER TO it185158;

```

Το function αυτό δέχεται ως παραμέτρους τα στοιχεία του πελάτη, αντίστοιχα με τα columns στον πίνακα guests, το επιθυμητό είδος δωματίου, το οποίο δίνεται σε μορφή integer στον κώδικα, και τις ημερομηνίες για την κράτηση. Στον κώδικα γίνεται έλεγχος, αν ο πελάτης υπάρχει στην βάση δεδομένων με βάση την ταυτότητα του (guest_id), που σημαίνει ότι έχει κάνει κι άλλη κράτηση στο παρελθόν, και στην περίπτωση που υπάρχει, γίνεται επικαιροποίηση των στοιχείων του στον πίνακα guests. Στην περίπτωση που δεν υπήρχε, δημιουργείται νέα σειρά στον πίνακα guests.

Έπειτα, γίνεται τυχαία επιλογή κάποιου δωματίου από τον πίνακα rooms με βάση την παράμετρο του είδους δωματίου και γίνεται έλεγχος στον πίνακα reserved_rooms για το αν υπάρχει reservation για αυτό το δωμάτιο που συμπίπτει με τις ημερομηνίες του νέου reservation. Στην περίπτωση που δεν υπάρχει κανένα διαθέσιμο δωμάτιο του πίνακα rooms, για εκείνες τις ημερομηνίες δεν δημιουργείται το reservation.

```

CREATE OR REPLACE FUNCTION pastreservation(reservation_id integer)
RETURNS void AS
$BODY$
BEGIN
    INSERT INTO past_reservations (idpastreservation, room_id, guest_id, since_when,
check_out)

```

```

SELECT idreservation, room_id, guest_id, since_when, check_out FROM reserved_rooms
WHERE idreservation= $1;
END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION pastreservation(integer)
OWNER TO it185158;

```

Το Function αυτό δέχεται ως παράμετρο το μοναδικό id ενός reservation από τον πίνακα reserved_rooms και δημιουργεί ένα αντίγραφο της σειράς αυτής στον πίνακα past_reservations.

```

CREATE OR REPLACE FUNCTION process_log_table()
RETURNS trigger AS
$BODY$
BEGIN
IF (TG_OP = 'DELETE') THEN
INSERT INTO log_table SELECT 'D', now(), user, OLD.*;
RETURN OLD;
ELSIF (TG_OP = 'UPDATE') THEN
INSERT INTO log_table SELECT 'U', now(), user, NEW.*;
RETURN NEW;
ELSIF (TG_OP = 'INSERT') THEN
INSERT INTO log_table SELECT 'I', now(), user, NEW.*;
RETURN NEW;
END IF;
RETURN NULL;
END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION process_log_table()
OWNER TO it185158;

```

Το συγκεκριμένο trigger function εισάγει τα κατάλληλα δεδομένα στον πίνακα log_table ο οποίος εξυπηρετεί ως log files. Το function αυτό εκτελείται μετά από οποιαδήποτε ενέργεια εισαγωγής, επικαιροποίησης, ή διαγραφής στον πίνακα reserved_rooms.

```

CREATE TRIGGER log_table
AFTER INSERT OR UPDATE OR DELETE
ON reserved_rooms
FOR EACH ROW
EXECUTE PROCEDURE process_log_table();

```

Το trigger αυτό καλεί την διαδικασία process_log_table().

3.2. ΑΝΑΖΗΤΗΣΗ

Οι παρακάτω διαδικασίες εκτελούν αναζήτηση και εμφάνιση των στοιχείων των πινάκων.

```
CREATE OR REPLACE FUNCTION getlogfile()
  RETURNS SETOF log_table AS
$BODY$
BEGIN
  RETURN QUERY
  SELECT * FROM log_table;
END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100
ROWS 1000;
ALTER FUNCTION getlogfile()
  OWNER TO it185158;
```

Η διαδικασία αυτή επιστρέφει ολόκληρο τον πίνακα log_table.

```
CREATE OR REPLACE FUNCTION getreservations(IN guest_id integer)
  RETURNS TABLE(guest_name character varying, guest_surname character varying,
guest_mobile text, room_id integer, checkin text, checkout text) AS
$BODY$
BEGIN
  RETURN QUERY
  SELECT
    g.guest_name,
    g.guest_surname,
    g.guest_mobile,
    r.room_id,
    r.since_when,
    r.check_out
  FROM
    reserved_rooms r
  INNER JOIN guests g ON r.guest_id = g.idguest
  WHERE
    r.guest_id = $1;

END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100
ROWS 1000;
ALTER FUNCTION getreservations(integer)
  OWNER TO it185158;
```

Η διαδικασία αυτή αναζητά μια σειρά στον πίνακα reserved_rooms με βάση το ID του πελάτη, και επιστρέφει τα στοιχεία του, μαζί με τα στοιχεία της κράτησης του.

```
CREATE OR REPLACE FUNCTION searchdate(IN searched_date text)
```

```

    RETURNS TABLE(reservation_id integer, room_id integer, guest_id integer, since_when text,
    check_out text) AS
$BODY$
    SELECT idreservation, room_id, guest_id, since_when, check_out
    FROM reserved_rooms
    WHERE since_when = $1;
$BODY$
LANGUAGE sql VOLATILE
COST 100
ROWS 1000;
ALTER FUNCTION searchdate(text)
OWNER TO it185158;

```

Η παραπάνω διαδικασία δέχεται ως παράμετρο μια ημερομηνία, και αναζητά τις κρατήσεις στον πίνακα reserved_rooms, με βάση την ημερομηνία έναρξης της διαμονής, η οποία διαβάζεται από την παράμετρο.

```

CREATE OR REPLACE FUNCTION searchid(IN searched_id integer)
    RETURNS TABLE(idres integer, idguest integer, roomid integer, since_when text,
    check_out text) AS
$BODY$
    SELECT idreservation, room_id, guest_id, since_when, check_out
    FROM reserved_rooms
    WHERE guest_id = $1;
$BODY$
LANGUAGE sql VOLATILE
COST 100
ROWS 1000;
ALTER FUNCTION searchid(integer)
OWNER TO it185158;

```

Η παραπάνω διαδικασία, δέχεται ως παράμετρο έναν integer που αντιστοιχεί στο ID ενός πελάτη, και κάνει αναζήτηση στον πίνακα reserved_rooms για τις κρατήσεις που υπάρχουν για τον πελάτη με αυτό το ID.

```

CREATE OR REPLACE FUNCTION searchroom(IN searched_room integer)
    RETURNS TABLE(reservation_id integer, room_id integer, guest_id integer, since_when
    text, check_out text) AS
$BODY$
    SELECT idreservation, room_id, guest_id, since_when, check_out
    FROM reserved_rooms
    WHERE room_id = $1;
$BODY$
LANGUAGE sql VOLATILE
COST 100
ROWS 1000;
ALTER FUNCTION searchroom(integer)

```



```
OWNER TO it185158;
```

Η παραπάνω διαδικασία δέχεται ως παράμετρο έναν integer που αντιστοιχεί στον αριθμό δωματίου, και εμφανίζει όλες τις κρατήσεις που έχουν γίνει για αυτό το δωμάτιο από τον πίνακα reserved_rooms.

3.3. ΕΠΙΚΑΙΡΟΠΟΙΗΣΗ

Η παρακάτω διαδικασία εκτελεί επικαιροποίηση των πινάκων.

```
CREATE OR REPLACE FUNCTION updatereservation(  
  g_id integer,  
  new_g_name character varying,  
  new_g_surname character varying,  
  new_g_mobile text,  
  new_room_type integer,  
  new_checkin text,  
  new_checkout text)  
RETURNS void AS  
$BODY$  
DECLARE  
  temp_room INT;  
  current_category INT;  
BEGIN  
  SELECT r.roomcategory INTO current_category  
  FROM reserved_rooms rr  
  JOIN rooms r ON rr.room_id = r.idroom  
  WHERE rr.guest_id = $1;  
  
  UPDATE guests  
  SET  
    guest_name = new_g_name,  
    guest_surname = new_g_surname,  
    guest_mobile = new_g_mobile  
  WHERE idguest = $1;  
  
  IF current_category IS NULL OR (current_category IS NOT NULL AND current_category  
  != new_room_type) THEN  
    SELECT r.idroom INTO temp_room  
    FROM rooms r  
    LEFT JOIN reserved_rooms rr ON r.idroom = rr.room_id  
    WHERE r.roomcategory = new_room_type  
    AND rr.room_id IS NULL  
    AND NOT EXISTS (  
      SELECT 1  
      FROM reserved_rooms
```

```

        WHERE r.idroom = room_id
        AND to_timestamp(since_when, 'DD-MM-YYYY') <=
to_timestamp(new_checkout, 'DD-MM-YYYY')
        AND to_timestamp(check_out, 'DD-MM-YYYY') >= to_timestamp(new_checkin,
'DD-MM-YYYY')
    )
    ORDER BY random()
    LIMIT 1;

    IF temp_room IS NOT NULL THEN
        UPDATE reserved_rooms
        SET
            room_id = temp_room,
            since_when = new_checkin,
            check_out = new_checkout
        WHERE guest_id = $1;
    ELSE
        RAISE EXCEPTION 'No available room for reservation.';
    END IF;
ELSE
    UPDATE reserved_rooms
    SET
        since_when = new_checkin,
        check_out = new_checkout
    WHERE guest_id = $1;
END IF;
END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION updatereservation(integer, character varying, character varying, text,
integer, text, text)
OWNER TO it185158;

```

Η διαδικασία αυτή δέχεται ως παράμετρο το ID ενός πελάτη και τα νέα στοιχεία της κράτησης, και κάνει update την κράτηση με βάση το ID του χρήστη. Αλλάζει τα στοιχεία του χρήστη, και τις ημερομηνίες κράτησης, και στην περίπτωση που γίνει αλλαγή στην κατηγορία δωματίου, αλλάζει και το δωμάτιο.

3.4. ΔΙΑΓΡΑΦΗ

Οι παρακάτω διαδικασίες εκτελούν διαγραφή κάποιας σειράς από πίνακα.

```

CREATE OR REPLACE FUNCTION reservationcanceled(reservation integer)
RETURNS void AS

```

```

$BODY$
BEGIN
    DELETE FROM reserved_rooms WHERE idreservation = $1;
END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION reservationcanceled(integer)
OWNER TO it185158;

```

Η παραπάνω διαδικασία διαγράφει μια σειρά του πίνακα `reserved_rooms`, με βάση την παράμετρο που αντιστοιχεί σε κάποιο `idreservation`, στην περίπτωση ακύρωσης μιας κράτησης.

```

CREATE OR REPLACE FUNCTION reservationcompleted()
RETURNS trigger AS
$BODY$
BEGIN
    DELETE FROM reserved_rooms WHERE idreservation = NEW.idpastreservation;
    RETURN NEW;
END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION reservationcompleted()
OWNER TO it185158;

```

Η παραπάνω είναι μια `trigger function` που διαγράφει μια σειρά από τον πίνακα `reserved_rooms` όταν αυτή έχει αντιγραφεί στον πίνακα `past_reservations`. Κάτι τέτοιο συμβαίνει όταν μια κράτηση έχει ολοκληρωθεί.

```

CREATE TRIGGER past_trigger
AFTER INSERT
ON past_reservations
FOR EACH ROW
EXECUTE PROCEDURE reservationcompleted();

```

Το παραπάνω είναι το `trigger` που εκτελεί την διαδικασία `reservationcompleted()` μετά από την εισαγωγή μιας νέας σειράς στον πίνακα `past_reservations`.

4. ΚΩΔΙΚΑΣ ΔΙΕΠΑΦΗΣ

Ο κώδικας της διεπαφής έχει γραφτεί σε `Java Swing`, πάνω στο πρόγραμμα `NetBeans`. Αποτελείται από μία κλάση `main`, στην οποία έχουν γραφτεί οι μέθοδοι για τις λειτουργίες της διεπαφής, και μία κλάση `DateSelectionHandler` η οποία εμφανίζει τον πίνακα επιλογής ημερομηνίας για την δημιουργία μιας νέας κράτησης.

Για τη σύνδεση με την βάση, δίνονται τα απαραίτητα στοιχεία, με τον αντίστοιχο κώδικα, και καλείται το αντίστοιχο statement για την πραγματοποίηση της σύνδεσης με τη βάση, μέσα στην κάθε μέθοδο, πριν την εκτέλεση της επιθυμητής ενέργειας.

```
static String  driverClassName = "org.postgresql.Driver" ;
static String  url = "jdbc:postgresql://dblabs.iee.ihu.gr:5432/it185158" ;
static Connection dbConnection = null;
static String  username = "it185158";
static String  passwd = "tecvasded123!";
static Statement statement = null;
```

```
private void createActionPerformed(java.awt.event.ActionEvent evt) {

    try {
        int id_user = Integer.parseInt(IDno.getText());
        String u_name = onoma.getText();
        String u_lastname = epitheto.getText();
        String kinitoNo = mobile.getText();
        String comein = (String) checkin.getText();
        String leaves = (String) checkout.getText();
        int room_type;

        if(single.isSelected()){
            room_type = 1;
        }else if(dual.isSelected()){
            room_type = 2;
        }else{
            room_type = 3;
        }

        Class.forName (driverClassName);
        dbConnection = DriverManager.getConnection (url, username, passwd);
        statement = dbConnection.createStatement();
        statement.executeQuery("select  createreservation('\" + u_name + \"', '\" + u_lastname + \"', '\" + id_user + \"', '\" + kinitoNo + \"', '\" + room_type + \"', '\" + comein + \"', '\" + leaves + \"')");
        System.out.println("Reservation Created Successfully");
        statement.close();
        dbConnection.close();
    } catch (ClassNotFoundException | SQLException ex) {
        Logger.getLogger(Hotel.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

Ο παραπάνω κώδικας, είναι ένα παράδειγμα, αφορά την δημιουργία μιας νέας κράτησης. Βρίσκεται στην main κλάση της διεπαφής, δέχεται τα στοιχεία για τις μεταβλητές των παραμέτρων μέσα από το text box και radio buttons της οθόνης Check IN. Γίνεται σύνδεση με την βάση μέσα στη μέθοδο, και καλείται η αντίστοιχη διαδικασία.

Αντίστοιχες μέθοδοι υπάρχουν και για την επικαιροποίηση στοιχείων, τη διαγραφή, και την αναζήτηση.

- SearchButtonActionPerformed() Η μέθοδος κάνει αναζήτηση κελιών, η οποία με ένα if κάνει αναζήτηση με βάση το Guest ID, Room ID, ή Dates, και καλεί τις αντίστοιχες μεθόδους searchid(), searchroom() και searchdates() της βάσης.
- CheckOutActionPerformed() Η μέθοδος η οποία καλείται όταν γίνεται επιλογή από την οθόνη Reservations της διεπαφής μιας κράτησης, και με το πάτημα του κουμπιού checkout. Η μέθοδος καλεί την διαδικασία pastreservation() της βάσης. Αφορά τις κρατήσεις που έχουν ολοκληρωθεί.
- UpdateActionPerformed() είναι η μέθοδος που διαχειρίζεται την επικαιροποίηση, παίρνοντας τα στοιχεία για τις μεταβλητές των παραμέτρων από τα text boxes και radio buttons της οθόνης Update της διεπαφής, και καλεί την διαδικασία updatereservation() της βάσης.
- createActionPerformed() είναι η μέθοδος που διαχειρίζεται την εισαγωγή μιας νέας κράτησης. Παίρνει τις τιμές των μεταβλητών για τις παραμέτρους από τα text boxes και radio buttons της οθόνης Check IN της διεπαφής. Καλεί την μέθοδο createreservation() της βάσης.
- resultsMouseClicked() είναι η μέθοδος που με την επιλογή μιας κράτησης από της οθόνη Reservations της διεπαφής, μεταφέρει τα στοιχεία στην οθόνη Update της διεπαφής, ώστε να γίνει η επικαιροποίηση, σε όποιο στοιχείο χρειάζεται. Καλείται η μέθοδος getreservations της βάσης.
- cancelActionPerformed() Είναι η μέθοδος που διαγράφει μια κράτηση, με το πάτημα του κουμπιού cancel της οθόνης Reservations της διεπαφής, μετά την επιλογή μιας κράτησης στην οθόνη. Καλείται η μέθοδος reservationcanceled() της βάσης.
- logActionPerformed() είναι η μέθοδος που εμφανίζει το log file στην οθόνη Log της διεπαφής με την κλήση της μεθόδου getlogfile() της βάσης.

5. ΔΙΕΠΑΦΗ

Η διεπαφή χρήστη αποτελείται από 4 οθόνες. Η εναλλαγή οθονών γίνεται από τα tabs που υπάρχουν στην κορυφή της οθόνης. Οι οθόνες είναι οι παρακάτω:

- Check IN είναι η οθόνη στην οποία συμπληρώνονται τα στοιχεία για την δημιουργία μιας νέας κράτησης στα text boxes, radio buttons και pop ups για τις ημερομηνίες.

- Reservations είναι η οθόνη στην οποία γίνεται η αναζήτηση μιας κράτησης με βάση το Guest ID, την ημερομηνία check in, ή τον αριθμό δωματίου. Σε αυτή την οθόνη γίνεται η ολοκλήρωση και ακύρωση μιας κράτησης, ή η επιλογή μιας κράτησης για επικαιροποίηση.
- Update είναι η οθόνη στην οποία εμφανίζονται τα τρέχοντα στοιχεία της κράτησης που έγινε επιλογή στην οθόνη Reservations, και μπορούν να αλλαχθούν και να επικαιροποιηθούν.
- Log είναι η οθόνη στην οποία εμφανίζεται το log file, μια λίστα με όλες τις ενέργειες που εκτελέστηκαν. Η εμφάνιση γίνεται με το πάτημα του κουμπιού Show Log και με κάθε πάτημα γίνεται ανανέωση.