|  |
| --- |
| Γιωργος Λεωνιδης | Γαρυφαλλια-Μαρια Ευρυδογλου | Κωνσταντίνος Γράτσος |



|  |  |
| --- | --- |
| Τεχνολογια βασεων δεδομενων | Ανάπτυξη Πληροφοριακού συστήματος μηχανογράφησης ξενοδοχείου |

# Εισαγωγή

Στόχος της παρούσας εργασίας ήταν η πρακτική εξάσκηση και η εξοικείωση μας με τις έννοιες που διδάσκονται στο μάθημα, καθώς και η απόκτηση ουσιαστικής εμπειρίας από τη συνεργασία μας από απόσταση η οποία αποδείχτηκε πρόκληση για όλους μας.

Η σχεσιακή βάση δεδομένων αναπτύχθηκε σε PostgreSql. Οι τεχνολογίες και τα προγράμματα που χρησιμοποιήθηκαν απαριθμούνται συνοπτικά παρακάτω:

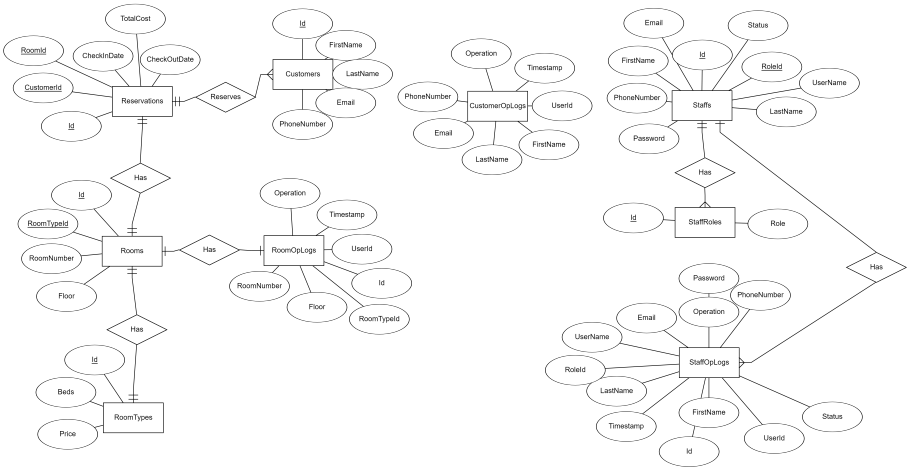
* PostgreSql (PgAdmin III)
* JAVA και JAVAFX (IntelliJ, SceneBuilder)
* GitHub .
* DB Visualizer

# Σχεδίαση σχεσιακής Βάσης Δεδομένων των ξενοδοχείου

Παρακάτω παρατίθεται το Relational Schema της βάσης δεδομένων του ξενοδοχείου:

# 

Και το ER Diagram:



# Σχεδίαση UI

Για την σχεδίαση της διεπαφής χρήστη χρησιμοποιήθηκε το εργαλείο SceneBuilder.

Τα Scenes αποτελούνται από:

* To Login,
* Το Interface του receptionist,

1. την Αρχική Σελίδα ,
2. την Νέα Κράτηση ,
3. την Διαγραφή Κράτησης,
4. την Ενημέρωση Κράτησης,
5. την Ενημέρωση Πελατολόγιου
6. τα Δυναμικά Ερωτήματα

* το Interface του διαχειριστή,

1. την Εισαγωγή Νέου Δωματίου
2. την Ενημέρωση Δωματίου
3. την Διαγραφή Δωματίου
4. την Διαγραφή Πελάτη
5. την Εισαγωγή Νέου Υπαλλήλου
6. την Ενημέρωση Υπαλλήλου
7. την Διαγραφή Υπαλλήλου
8. και τα Log Files

## Παράρτημα Α

PL/pgSQL functions

Functions στο πίνακα Rooms:

Αναζήτηση δωματίου με αριθμό (επιστρέφει όροφο, αριθμό δωματίου και τα κρεβάτια του δωματίου)

CREATE OR REPLACE FUNCTION findroom (RoomNum INT)

    RETURNS TABLE (

        Floor INT,

        RoomNumber INT,

        Beds INT

)

AS $$

BEGIN

    RETURN QUERY SELECT r."RoomNumber", r."Floor", r0."Beds"

      FROM "Rooms" AS r

      LEFT JOIN "RoomTypes" AS r0 ON r."RoomTypeId" = r0."Id"

      WHERE r."RoomNumber" = RoomNum

      LIMIT 1;

END; $$

LANGUAGE 'plpgsql';

Προσθήκη δωματίου

CREATE OR REPLACE FUNCTION addroom (RoomFloor INT, RoomNumber INT, RoomTypeId BIGINT)

    RETURNS TABLE (

        Id BIGINT

)

AS $$

BEGIN

    RETURN QUERY INSERT INTO "Rooms" ("Floor", "RoomNumber", "RoomTypeId")

      VALUES (RoomFloor, RoomNumber, RoomTypeId)

      RETURNING "Id";

END; $$

LANGUAGE 'plpgsql';

Ενημέρωση δωματίου

CREATE OR REPLACE FUNCTION updateroom (RoomId BIGINT, FloorNumber INT, RoomNumber INT

, RoomTypeId BIGINT)

    RETURNS void

AS $$

BEGIN

    UPDATE "Rooms" SET "Floor" = FloorNumber, "RoomNumber" = RoomNumber, "RoomTypeId" = RoomTypeId

      WHERE "Id" = RoomId;

END; $$

LANGUAGE 'plpgsql';

Αναζήτηση ID δωματίου με βάση τον αριθμό του

CREATE OR REPLACE FUNCTION getroomid (RoomNumber INT)

    RETURNS TABLE (

        Id BIGINT

)

AS $$

BEGIN

    RETURN QUERY  SELECT r."Id"

      FROM "Rooms" AS r

      WHERE r."RoomNumber" = RoomNumber

      LIMIT 1;

END; $$

LANGUAGE 'plpgsql';

Διαγραφή δωματίου

CREATE OR REPLACE FUNCTION deleteroom (RoomId BIGINT)

    RETURNS void

AS $$

BEGIN

    DELETE FROM "Rooms"

      WHERE "Id" = RoomId;

END; $$

LANGUAGE 'plpgsql';

Εμφάνιση όλων των δωματίων

CREATE OR REPLACE FUNCTION getrooms ()

    RETURNS TABLE (

        RoomId BIGINT,

     RoomNumber INT,

        Floor INT,

        Beds INT,

     RoomTypeId BIGINT

)

AS $$

BEGIN

    RETURN QUERY  SELECT r."Id",r."RoomNumber", r."Floor", r0."Beds",r."RoomTypeId"

      FROM "Rooms" AS r

      LEFT JOIN "RoomTypes" AS r0 ON r."RoomTypeId" = r0."Id";

END; $$

LANGUAGE 'plpgsql';

Εφάνιση όλων των ελεύθερων δωματίων σε μια συγκεκριμενή χρονική περίοδο

CREATE OR REPLACE FUNCTION getallfreerooms (CheckOutDate DATE, CheckInDate DATE)

    RETURNS TABLE (

        RoomId BIGINT,

        FloorNumber INT,

        RoomNumber INT,

        Beds INT

)

AS $$

BEGIN

    RETURN QUERY SELECT r."Id",r."Floor",r."RoomNumber",rt."Beds"

     FROM "Rooms" r JOIN "RoomTypes" rt on rt."Id"=r."RoomTypeId"

     WHERE r."Id" NOT IN

         (SELECT r0."Id"

          FROM "Reservations" AS r

          LEFT JOIN "Rooms" AS r0 ON r."RoomId" = r0."Id"

           WHERE not((r."CheckOutDate"<CheckInDate ::date) OR (r."CheckInDate">CheckOutDate ::date)));

END; $$

LANGUAGE 'plpgsql';

Αναζήτηση δωματίου με αριθμό (χρησιμευει για την ενημερωση δωματίου στον client)

CREATE OR REPLACE FUNCTION findroomforupdate (RoomNum INT)

    RETURNS TABLE (

        RoomId BIGINT,

        FloorNumber INT,

        RoomNumber INT,

        RoomTypeId BIGINT

)

AS $$

BEGIN

    RETURN QUERY SELECT r."Id" AS "RoomId", r."Floor" AS "FloorNumber", r."RoomNumber", r0."Id" AS "RoomTypeId"

      FROM "Rooms" AS r

      LEFT JOIN "RoomTypes" AS r0 ON r."RoomTypeId" = r0."Id"

      WHERE r."RoomNumber" = RoomNum

      LIMIT 1;

END; $$

LANGUAGE 'plpgsql';

Έλεγχος αν ενα δωμάτιο ειναι κλεισμένο γενικά

CREATE OR REPLACE FUNCTION checkroomstatus (RoomId BIGINT)

    RETURNS TABLE (

        ReservationId BIGINT,

        RoomNumber INT,

        firstName TEXT,

        lastName TEXT,

        checkInDate Date,

        checkOutDate Date,

        totalCost DECIMAL

)

AS $$

BEGIN

    RETURN QUERY SELECT r."Id", r0."RoomNumber", c."FirstName", c."LastName", r."CheckInDate"::date, r."CheckOutDate"::date, r."TotalCost"

      FROM "Reservations" AS r

      LEFT JOIN "Rooms" AS r0 ON r."RoomId" = r0."Id"

      LEFT JOIN "Customers" AS c ON r."CustomerId" = c."Id"

      WHERE r0."Id" = RoomId;

END; $$

LANGUAGE 'plpgsql';

Έλεγχος εαν ενα δωμάτιο είναι κλεισμένο αυτη τη στιγμή

CREATE OR REPLACE FUNCTION checkcurrentroomstatus (RoomNum INT)

    RETURNS TABLE (

        ReservationId BIGINT,

        RoomNumber INT,

        firstName TEXT,

        lastName TEXT,

        checkInDate Date,

        checkOutDate Date,

        cost DECIMAL

)

AS $$

BEGIN

    RETURN QUERY SELECT r."Id", r0."RoomNumber", c."FirstName", c."LastName", r."CheckInDate"::date, r."CheckOutDate"::date, r."TotalCost"

      FROM "Reservations" AS r

      LEFT JOIN "Rooms" AS r0 ON r."RoomId" = r0."Id"

      LEFT JOIN "Customers" AS c ON r."CustomerId" = c."Id"

      WHERE (r0."RoomNumber" = RoomNum) AND ((r."CheckInDate" <= current\_date) AND (r."CheckOutDate" >= current\_date));

END; $$

LANGUAGE 'plpgsql';

Επιστρέφει το κόστος ενος δωματίου

CREATE OR REPLACE FUNCTION checkroomcost (RoomNum INT)

    RETURNS TABLE (

        cost DECIMAL

)

AS $$

BEGIN

    RETURN QUERY SELECT r0."Price"

      FROM "Rooms" AS r

      LEFT JOIN "RoomTypes" AS r0 ON r."RoomTypeId" = r0."Id"

      WHERE r."RoomNumber" = RoomNum

      LIMIT 1;

END; $$

LANGUAGE 'plpgsql';

Επιστρέφει το πληθος ολων των check in ενός δωματίου

CREATE OR REPLACE FUNCTION totalroomcheckins(RoomNum INT)

    RETURNS TABLE (

        totalcheckins INT

)

AS $$

BEGIN

    RETURN QUERY SELECT COUNT(\*)::INT

      FROM "ReservationOperationsLogs" AS r

      LEFT JOIN "Rooms" AS r0 ON r."RoomId" = r0."Id"

      WHERE (r."Operation" = 'I') AND (r0."RoomNumber" = RoomNum);

END; $$

LANGUAGE 'plpgsql';

Functions στο πίνακα Customers:

Αναζήτηση πελάτη με όνομα και επίθετο

CREATE OR REPLACE FUNCTION searchcustomer (CustomerLastname TEXT, CustomerFirstName TEXT)

    RETURNS TABLE (

        Id BIGINT,

        Email TEXT,

        FirstName TEXT,

        LastName TEXT,

        PhoneNumber BIGINT

)

AS $$

BEGIN

    RETURN QUERY  SELECT c."Id", c."Email", c."FirstName", c."LastName", c."PhoneNumber"

      FROM "Customers" AS c

      WHERE ((upper(CustomerFirstName) = '')

      OR (strpos(upper(c."FirstName"), upper(CustomerFirstName)) > 0))

      AND ((upper(CustomerLastname) = '')

      OR (strpos(upper(c."LastName"), upper(CustomerLastname)) > 0));

END; $$

LANGUAGE 'plpgsql';

Προσθήκη καινούργιου πελάτη

CREATE OR REPLACE FUNCTION addcustomer (Email TEXT, FirstName TEXT, LastName TEXT,

PhoneNumber BIGINT)

    RETURNS TABLE (

        Id BIGINT

)

AS $$

BEGIN

    RETURN QUERY  INSERT INTO "Customers" ("Email", "FirstName", "LastName", "PhoneNumber")

      VALUES (Email, FirstName,LastName,PhoneNumber )

      RETURNING "Id";

END; $$

LANGUAGE 'plpgsql';

Επιστρέφει όλους τους πελάτες της βάσης

CREATE OR REPLACE FUNCTION getcustomers()

    RETURNS TABLE (

        Id BIGINT,

        email TEXT,

        firstName TEXT,

        lastName TEXT,

        phoneNumber BIGINT

)

AS $$

BEGIN

    RETURN QUERY  SELECT c."Id", c."Email", c."FirstName", c."LastName", c."PhoneNumber"

      FROM "Customers" AS c;

END; $$

LANGUAGE 'plpgsql';

Ενημέρωση ήδη υπάρχοντος πελάτη

CREATE OR REPLACE FUNCTION updateCustomer (CustomerId BIGINT,

CustomerFirstName TEXT,CustomerLastName TEXT,CustomerEmail TEXT,CustomerPhoneNumber BIGINT)

    RETURNS void

AS $$

BEGIN

    UPDATE "Customers" SET "Email" = CustomerEmail, "FirstName" = CustomerFirstName, "LastName" = CustomerLastName, "PhoneNumber" = CustomerPhoneNumber

      WHERE "Id" = CustomerId;

END; $$

LANGUAGE 'plpgsql';

Επιστρέφει το ID του πελάτη

CREATE OR REPLACE FUNCTION getcustomerid (CustomerLastName TEXT, CustomerFirstName TEXT)

    RETURNS TABLE (

        Id BIGINT

)

AS $$

BEGIN

    RETURN QUERY  SELECT c."Id"

      FROM "Customers" AS c

      WHERE (upper(c."FirstName") = upper(CustomerFirstName))

      AND (upper(c."LastName") = upper(CustomerLastName))

      LIMIT 1;

END; $$

LANGUAGE 'plpgsql';

Διαγραφή πελάτη

CREATE OR REPLACE FUNCTION deletecustomer (CustomerId BIGINT)

    RETURNS void

AS $$

BEGIN

    DELETE FROM "Customers"

      WHERE "Id" = CustomerId;

END; $$

LANGUAGE 'plpgsql';

Επιστρέφει τον αριθμό όλων των check in ενος πελάτη

CREATE OR REPLACE FUNCTION totalcustomercheckins(FirstName TEXT,LastName TEXT)

    RETURNS TABLE (

        totalcheckins INT

)

AS $$

BEGIN

    RETURN QUERY SELECT COUNT(\*)::INT

      FROM "ReservationOperationsLogs" AS r

      LEFT JOIN "Customers" AS c ON r."CustomerId" = c."Id"

      WHERE (r."Operation" = 'I') AND ((upper(c."FirstName") = upper(FirstName)) AND (upper(c."LastName") =upper(LastName)));

END; $$

LANGUAGE 'plpgsql';

Έλεγχος αν ενας πελάτης έχει δωμάτιο κλεισμένο αυτή τη στιγμή

CREATE OR REPLACE FUNCTION checkcurrentcustomerstatus (fName TEXT,lName TEXT)

    RETURNS TABLE (

        ReservationId BIGINT,

        RoomNumber INT,

        firstName TEXT,

        lastName TEXT,

        checkInDate Date,

        checkOutDate Date,

        cost DECIMAL

)

AS $$

BEGIN

    RETURN QUERY SELECT r."Id", r0."RoomNumber", c."FirstName", c."LastName", r."CheckInDate"::date, r."CheckOutDate"::date, r."TotalCost"

      FROM "Reservations" AS r

      LEFT JOIN "Rooms" AS r0 ON r."RoomId" = r0."Id"

      LEFT JOIN "Customers" AS c ON r."CustomerId" = c."Id"

      WHERE (c."FirstName" = fName) AND (c."LastName" = lName) AND ((r."CheckInDate" <= current\_date) AND (r."CheckOutDate" >= current\_date));

END; $$

LANGUAGE 'plpgsql';

Functions στο πίνακα Reservations:

Προσθήκη νέας κράτησης

CREATE OR REPLACE FUNCTION addreservation (CheckInDate DATE,

CheckOutDate DATE, CustomerId BIGINT,RoomId BIGINT,TotalCost DECIMAL)

    RETURNS TABLE (

        Id BIGINT

)

AS $$

BEGIN

    RETURN QUERY INSERT INTO "Reservations" ("CheckInDate", "CheckOutDate", "CustomerId", "RoomId", "TotalCost")

      VALUES (CheckInDate ,CheckOutDate , CustomerId , RoomId , TotalCost)

      RETURNING "Id";

END; $$

LANGUAGE 'plpgsql';

Αναζήτηση κρατήσεων σε μια συγκεκριμενη χρονική περίοδο

CREATE OR REPLACE FUNCTION searchreservation(CheckInDate DATE, CheckOutDate DATE)

    RETURNS TABLE (

        roomNumber INT,

        firstName TEXT,

        lastName TEXT

)

AS $$

BEGIN

    RETURN QUERY  SELECT r0."RoomNumber", c."FirstName", c."LastName"

      FROM "Reservations" AS r

      LEFT JOIN "Rooms" AS r0 ON r."RoomId" = r0."Id"

      LEFT JOIN "Customers" AS c ON r."CustomerId" = c."Id"

      WHERE not((r."CheckOutDate"< CheckInDate::date) OR (r."CheckInDate"> CheckOutDate::date));

END; $$

LANGUAGE 'plpgsql';

Επιστρέφει όλες τις υπάρχουσες κρατήσεις

CREATE OR REPLACE FUNCTION getreservations()

    RETURNS TABLE (

        Id BIGINT,

        roomNumber INT,

        firstName TEXT,

        lastName TEXT,

        checkInDate DATE

        checkOutDate DATE

        totalCost DECIMAL

)

AS $$

BEGIN

    RETURN QUERY SELECT r."Id", r0."RoomNumber", c."FirstName", c."LastName", r."CheckInDate"::date, r."CheckOutDate"::date, r."TotalCost"

      FROM "Reservations" AS r

      LEFT JOIN "Rooms" AS r0 ON r."RoomId" = r0."Id"

      LEFT JOIN "Customers" AS c ON r."CustomerId" = c."Id";

END; $$

LANGUAGE 'plpgsql';

Αναζήτηση κράτησης με στοιχεία πελάτη. Χρησιμοποιούμε την function της PostgreSQL strpos() που ελέγχει αν το substring υπάρχει (παρομοιο με το LIKE).

CREATE OR REPLACE FUNCTION searchspecificreservation (CustomerLastName TEXT,

CustomerFirstName TEXT)

    RETURNS TABLE (

        ReservationId BIGINT,

        RoomNumber INT,

        firstName TEXT,

        lastName TEXT,

        checkInDate Date,

        checkOutDate Date,

        totalCost DECIMAL

)

AS $$

BEGIN

    RETURN QUERY SELECT r."Id", r0."RoomNumber", c."FirstName", c."LastName", r."CheckInDate"::Date, r."CheckOutDate"::Date, r."TotalCost"

      FROM "Reservations" AS r

      LEFT JOIN "Customers" AS c ON r."CustomerId" = c."Id"

      LEFT JOIN "Rooms" AS r0 ON r."RoomId" = r0."Id"

      WHERE ((upper(CustomerFirstName) = '')

      OR (strpos(upper(c."FirstName"),upper(CustomerFirstName)) > 0))

      AND ((upper(CustomerLastName) = '')

      OR (strpos(upper(c."LastName"), upper(CustomerLastName)) > 0));

END; $$

LANGUAGE 'plpgsql';

Διαγραφή κράτησης

CREATE OR REPLACE FUNCTION deletereservation (ReservationId BIGINT)

    RETURNS void

AS $$

BEGIN

    DELETE FROM "Reservations"

      WHERE "Id" =ReservationId;

END; $$

LANGUAGE 'plpgsql';

Ενημέρωση κράτησης

CREATE OR REPLACE FUNCTION updatereservation (ReservationId BIGINT, CheckInDate DATE,

CheckOutDate DATE,RoomId BIGINT,CustomerId BIGINT,TotalCost DECIMAL)

    RETURNS void

AS $$

BEGIN

    UPDATE "Reservations" SET "CheckInDate" = CheckInDate, "CheckOutDate" = CheckOutDate, "RoomId" = RoomId, "TotalCost" = TotalCost

      WHERE "Id" = ReservationId;

END; $$

LANGUAGE 'plpgsql';

Αναζήτηση κράτηση με ID πελάτη

CREATE OR REPLACE FUNCTION searchspecificreservationwithid (CustomerId BIGINT)

    RETURNS TABLE (

        ReservationId BIGINT,

        RoomNumber INT,

        firstName TEXT,

        lastName TEXT,

        checkInDate TIMESTAMP,

        checkOutDate TIMESTAMP,

        totalCost DECIMAL

)

AS $$

BEGIN

    RETURN QUERY SELECT r."Id", r0."RoomNumber", c."FirstName", c."LastName", r."CheckInDate", r."CheckOutDate", r."TotalCost"

      FROM "Reservations" AS r

      LEFT JOIN "Customers" AS c ON r."CustomerId" = c."Id"

      LEFT JOIN "Rooms" AS r0 ON r."RoomId" = r0."Id"

      WHERE c."Id" = CustomerId;

END; $$

LANGUAGE 'plpgsql';

Επιστρέφει το σύνολο των εσόδων ανα μήνα

CREATE OR REPLACE FUNCTION getincomepermonth()

    RETURNS TABLE (

        checkingMonth TIMESTAMP,

        monthlySum DECIMAL

)

AS $$

BEGIN

      RETURN QUERY SELECT date\_trunc('month', "CheckInDate") AS checking\_month, sum("TotalCost") as monthly\_sum

     FROM "Reservations"

 GROUP BY checking\_month;

END; $$

LANGUAGE 'plpgsql';

Εμφάνιση δωματίων με κόστος μικρότερο/μεγαλύτερο απο μια τιμή

CREATE OR REPLACE FUNCTION reservationcostless (TotalCost DECIMAL)

    RETURNS TABLE (

        ReservationId BIGINT,

        RoomNumber INT,

        firstName TEXT,

        lastName TEXT,

        checkInDate Date,

        checkOutDate Date,

        cost DECIMAL

)

AS $$

BEGIN

    RETURN QUERY SELECT r."Id", r0."RoomNumber", c."FirstName", c."LastName", r."CheckInDate"::date, r."CheckOutDate"::date, r."TotalCost"

      FROM "Reservations" AS r

      LEFT JOIN "Rooms" AS r0 ON r."RoomId" = r0."Id"

      LEFT JOIN "Customers" AS c ON r."CustomerId" = c."Id"

      WHERE r."TotalCost" < TotalCost;

END; $$

LANGUAGE 'plpgsql';

CREATE OR REPLACE FUNCTION reservationcostmore (TotalCost DECIMAL)

    RETURNS TABLE (

        ReservationId BIGINT,

        RoomNumber INT,

        firstName TEXT,

        lastName TEXT,

        checkInDate Date,

        checkOutDate Date,

        cost DECIMAL

)

AS $$

BEGIN

    RETURN QUERY SELECT r."Id", r0."RoomNumber", c."FirstName", c."LastName", r."CheckInDate"::date, r."CheckOutDate"::date, r."TotalCost"

      FROM "Reservations" AS r

      LEFT JOIN "Rooms" AS r0 ON r."RoomId" = r0."Id"

      LEFT JOIN "Customers" AS c ON r."CustomerId" = c."Id"

      WHERE r."TotalCost" > TotalCost;

END; $$

LANGUAGE 'plpgsql';

Επιστρέφει το πλήθος των κρατήσεων

CREATE OR REPLACE FUNCTION searchreservationcount(CheckInDate DATE, CheckOutDate DATE)

    RETURNS TABLE (

        reservations INT

)

AS $$

BEGIN

    RETURN QUERY SELECT COUNT(\*)::INT

      FROM "Reservations" AS r

      WHERE not((r."CheckOutDate"< CheckInDate::date) OR (r."CheckInDate"> CheckOutDate::date));

END; $$

LANGUAGE 'plpgsql';

Επιστρέφει τα έσοδα ενος συγκεκριμένου μήνα

CREATE OR REPLACE FUNCTION getspecificmonthincone(monthToCheck INT)

    RETURNS TABLE (

        checkingMonth TIMESTAMP,

        monthlySum DECIMAL

)

AS $$

BEGIN

      RETURN QUERY SELECT date\_trunc('month', "CheckInDate") AS checking\_month, sum("TotalCost") as monthly\_sum

     FROM "Reservations"

     WHERE date\_part('month',"CheckInDate")::INT = monthToCheck

      GROUP BY checking\_month;

END; $$

LANGUAGE 'plpgsql';

Functions στο πίνακα RoomTypes:

Επιστρέφει το κόστος του τύπου

CREATE OR REPLACE FUNCTION getroomtypeprice (Beds INT)

    RETURNS TABLE (

        Price DECIMAL

)

AS $$

BEGIN

    RETURN QUERY SELECT r."Price"

      FROM "RoomTypes" AS r

      WHERE r."Beds" = Beds

      LIMIT 1;

END; $$

LANGUAGE 'plpgsql';

Επιστρέφει τις συνολικές κρατήσεις ανα τύπο

CREATE OR REPLACE FUNCTION getreservedroomtypes ()

    RETURNS TABLE (

        TwoBeds INT,

        ThreeBeds INT,

        Suite INT

)

AS $$

BEGIN

    twoBeds := (SELECT COUNT(\*)::INT

      FROM "Reservations" AS r

      LEFT JOIN "Rooms" AS r0 ON r."RoomId" = r0."Id"

      LEFT JOIN "RoomTypes" AS r1 ON r0."RoomTypeId" = r1."Id"

      WHERE r1."Beds" = 2);

      threeBeds:= (SELECT COUNT(\*)::INT

      FROM "Reservations" AS r

      LEFT JOIN "Rooms" AS r0 ON r."RoomId" = r0."Id"

      LEFT JOIN "RoomTypes" AS r1 ON r0."RoomTypeId" = r1."Id"

      WHERE r1."Beds" = 3);

      suite:= (SELECT COUNT(\*)::INT

      FROM "Reservations" AS r

      LEFT JOIN "Rooms" AS r0 ON r."RoomId" = r0."Id"

      LEFT JOIN "RoomTypes" AS r1 ON r0."RoomTypeId" = r1."Id"

      WHERE r1."Beds" = 4);

      RETURN QUERY SELECT twoBeds,threeBeds,suite;

END; $$

LANGUAGE 'plpgsql';

Επιστρέφει το πλήθος των checkins ανα ημέρα

CREATE OR REPLACE FUNCTION checkinsbydate()

    RETURNS TABLE (

        checkInDate DATE,

        totalCheckIns BIGINT

)

AS $$

BEGIN

      RETURN QUERY select r."CheckInDate"::date,count(\*)

    from "Reservations" r

    group by r."CheckInDate";

END; $$

LANGUAGE 'plpgsql';

Επιστρέφει τις συνολικές κρατήσεις ανα τύπο (διαφορετική απο τη προηγουμενη function καθως αυτη μπορει να μην επιστρέψει εναν τυπο αν δεν υπαρχουν checkin).

CREATE OR REPLACE FUNCTION getgroupedreservedroomtypes()

    RETURNS TABLE (

        Beds INT,

        CheckIns BIGINT

)

AS $$

BEGIN

      RETURN QUERY select r0."Beds",count(\*)

from "Reservations" r join "Rooms" r1 on r."RoomId"=r1."Id" join "RoomTypes" r0 on r0."Id"=r1."RoomTypeId"

group by r0."Beds";

END; $$

LANGUAGE 'plpgsql';

Functions στο πίνακα Staffs:

Προσθήκη χρήστη

CREATE OR REPLACE FUNCTION addstaff (Email TEXT, FirstName TEXT, LastName TEXT,

PasswordText TEXT, PhoneNumber BIGINT,RoleId BIGINT, UserName TEXT)

    RETURNS TABLE (

        Id BIGINT

)

AS $$

BEGIN

    RETURN QUERY INSERT INTO "Staffs" ("Email", "FirstName", "LastName", "Password", "PhoneNumber", "RoleId", "UserName")

      VALUES (Email, FirstName, LastName, PasswordText, PhoneNumber, RoleId, UserName)

      RETURNING "Id";

END; $$

LANGUAGE 'plpgsql';

Έλεγχος ύπαρξης χρήστη

CREATE OR REPLACE FUNCTION checkstaff (UserName TEXT, PasswordText TEXT)

    RETURNS TABLE (

        Id BIGINT,

        FirstName TEXT,

        LastName TEXT,

        UserNameText TEXT,

        RoleText TEXT,

        PhoneNumber BIGINT,

        Email TEXT

)

AS $$

BEGIN

    RETURN QUERY SELECT s."Id", s."FirstName", s."LastName",s."UserName", s0."Role", s."PhoneNumber", s."Email"

      FROM "Staffs" AS s

      LEFT JOIN "StaffRoles" AS s0 ON s."RoleId" = s0."Id"

      WHERE (s."UserName" = UserName) AND (s."Password" = PasswordText)

      LIMIT 1;

END; $$

LANGUAGE 'plpgsql';

Επιστρέφει όλους τους χρήστες

CREATE OR REPLACE FUNCTION getallstaff ()

    RETURNS TABLE (

        Id BIGINT,

        FirstName TEXT,

        LastName TEXT,

        UserName TEXT,

        Password TEXT,

        RoleId BIGINT,

        PhoneNumber BIGINT,

        Email TEXT

)

AS $$

BEGIN

    RETURN QUERY SELECT s."Id", s."FirstName", s."LastName", s."UserName",s."Password",s."RoleId", s."PhoneNumber", s."Email"

      FROM "Staffs" AS s

      LEFT JOIN "StaffRoles" AS s0 ON s."RoleId" = s0."Id";

END; $$

LANGUAGE 'plpgsql';

Διαγραφή χρήστη

CREATE OR REPLACE FUNCTION deletestaff (StaffId BIGINT)

    RETURNS void

AS $$

BEGIN

    DELETE FROM "Staffs"

      WHERE "Id" = StaffId;

END; $$

LANGUAGE 'plpgsql';

Επιστρέφει το ID του ρόλου του χρήστη

CREATE OR REPLACE FUNCTION getstaffroleid (UserName TEXT, PasswordText TEXT)

    RETURNS TABLE (

        RoleId BIGINT

)

AS $$

BEGIN

    RETURN QUERY  SELECT s0."Id"

      FROM "Staffs" AS s

      LEFT JOIN "StaffRoles" AS s0 ON s."RoleId" = s0."Id"

      WHERE (s."UserName" = UserName) AND (s."Password" = PasswordText)

      LIMIT 1;

END; $$

LANGUAGE 'plpgsql';

Έλεγχος χρήστη με email

CREATE OR REPLACE FUNCTION checkstaffbymail (EmailText TEXT)

    RETURNS TABLE (

        Id BIGINT,

        FirstName TEXT,

        LastName TEXT,

        RoleText TEXT,

        PhoneNumber BIGINT,

        Email TEXT

)

AS $$

BEGIN

    RETURN QUERY SELECT s."Id", s."FirstName", s."LastName", s0."Role", s."PhoneNumber", s."Email"

      FROM "Staffs" AS s

      LEFT JOIN "StaffRoles" AS s0 ON s."RoleId" = s0."Id"

      WHERE s."Email" = EmailText;

END; $$

LANGUAGE 'plpgsql';

Έλεγχος χρήστη με όνομα/επίθετο

CREATE OR REPLACE FUNCTION checkstaffbyname (FirstNameText TEXT,LastNameText TEXT)

    RETURNS TABLE (

        Id BIGINT,

        FirstName TEXT,

        LastName TEXT,

        UserName TEXT,

        Password TEXT,

        RoleId BIGINT,

        PhoneNumber BIGINT,

        Email TEXT

)

AS $$

BEGIN

    RETURN QUERY SELECT s."Id", s."FirstName", s."LastName", s."UserName",s."Password",s."RoleId", s."PhoneNumber", s."Email"

      FROM "Staffs" AS s

      LEFT JOIN "StaffRoles" AS s0 ON s."RoleId" = s0."Id"

      WHERE (upper(s."FirstName") = upper(FirstNameText)) AND (upper(s."LastName") = upper(LastNameText));

END; $$

LANGUAGE 'plpgsql';

Έλεγχος χρήστη με username

CREATE OR REPLACE FUNCTION checkstaffbyusername (UserName TEXT)

    RETURNS TABLE (

        Id BIGINT,

        FirstName TEXT,

        LastName TEXT,

        RoleText TEXT,

        PhoneNumber BIGINT,

        Email TEXT

)

AS $$

BEGIN

    RETURN QUERY SELECT s."Id", s."FirstName", s."LastName", s0."Role", s."PhoneNumber", s."Email"

      FROM "Staffs" AS s

      LEFT JOIN "StaffRoles" AS s0 ON s."RoleId" = s0."Id"

      WHERE (upper(s."UserName") = upper(UserName));

END; $$

LANGUAGE 'plpgsql';

Ενημέρωση χρήστη

CREATE OR REPLACE FUNCTION updatestaff (StaffId BIGINT, FirstName TEXT,

LastName TEXT,UserName TEXT,Email TEXT,PhoneNumber BIGINT,RoleId BIGINT)

    RETURNS void

AS $$

BEGIN

    UPDATE "Staffs" SET "Email" = Email, "RoleId" = RoleId, "FirstName" = FirstName, "LastName" = LastName, "PhoneNumber" = PhoneNumber, "UserName" = UserName

      WHERE "Id" = StaffId;

END; $$

LANGUAGE 'plpgsql';

Είσοδος χρήστη στο σύστημα (προς το παρον ειναι σχεδιασμένο για ενα χρήστη τη φορά)

CREATE OR REPLACE FUNCTION signintstaff (StaffId BIGINT)

    RETURNS void

AS $$

BEGIN

    UPDATE "Staffs" SET "Status" = 1

      WHERE "Id" = StaffId;

END; $$

LANGUAGE 'plpgsql';

Έξοδος χρήστη απο το σύστημα

CREATE OR REPLACE FUNCTION signoutstaff ()

    RETURNS void

AS $$

BEGIN

    UPDATE "Staffs" SET "Status" = 0

      WHERE "Status" = 1;

END; $$

LANGUAGE 'plpgsql';

Functions στους operation πίνακες:

Δημιουργία πίνακα log και trigger για τα operations στον πίνακα Customers

CREATE OR REPLACE FUNCTION process\_customer\_operations()

RETURNS TRIGGER AS $$

    BEGIN

        IF (TG\_OP = 'DELETE') THEN

            INSERT INTO "CustomerOperationsLogs" SELECT 'D', now(),(SELECT s."UserName"

      FROM "Staffs" AS s

      WHERE s."Status" = 1

      LIMIT 1), OLD.\*;

            RETURN OLD;

        ELSIF (TG\_OP = 'UPDATE') THEN

            INSERT INTO "CustomerOperationsLogs" SELECT 'U', now(),( SELECT s."UserName"

      FROM "Staffs" AS s

      WHERE s."Status" = 1

      LIMIT 1), NEW.\*;

            RETURN NEW;

        ELSIF (TG\_OP = 'INSERT') THEN

            INSERT INTO "CustomerOperationsLogs" SELECT 'I', now(), (SELECT s."UserName"

      FROM "Staffs" AS s

      WHERE s."Status" = 1

      LIMIT 1), NEW.\*;

            RETURN NEW;

        END IF;

        RETURN NULL;

    END;

$$

LANGUAGE plpgsql;

CREATE TRIGGER customer\_audit

AFTER INSERT OR UPDATE OR DELETE ON "Customers"

    FOR EACH ROW EXECUTE PROCEDURE process\_customer\_operations();

Δημιουργία πίνακα log και trigger για τα operations στον πίνακα Rooms:

CREATE OR REPLACE FUNCTION process\_room\_operations()

RETURNS TRIGGER AS $$

    BEGIN

        IF (TG\_OP = 'DELETE') THEN

            INSERT INTO "RoomOperationsLogs" SELECT 'D', now(),(SELECT s."UserName"

      FROM "Staffs" AS s

      WHERE s."Status" = 1

      LIMIT 1), OLD.\*;

            RETURN OLD;

        ELSIF (TG\_OP = 'UPDATE') THEN

            INSERT INTO "RoomOperationsLogs" SELECT 'U', now(),( SELECT s."UserName"

      FROM "Staffs" AS s

      WHERE s."Status" = 1

      LIMIT 1), NEW.\*;

            RETURN NEW;

        ELSIF (TG\_OP = 'INSERT') THEN

            INSERT INTO "RoomOperationsLogs" SELECT 'I', now(), (SELECT s."UserName"

      FROM "Staffs" AS s

      WHERE s."Status" = 1

      LIMIT 1), NEW.\*;

            RETURN NEW;

        END IF;

        RETURN NULL;

    END;

$$

LANGUAGE plpgsql;

CREATE TRIGGER room\_audit

AFTER INSERT OR UPDATE OR DELETE ON "Rooms"

    FOR EACH ROW EXECUTE PROCEDURE process\_room\_operations();

Δημιουργία πίνακα log και trigger για τα operations στον πίνακα Reservations :

CREATE OR REPLACE FUNCTION process\_reservation\_operations()

RETURNS TRIGGER AS $$

    BEGIN

        IF (TG\_OP = 'DELETE') THEN

            INSERT INTO "ReservationOperationsLogs" SELECT 'D', now(),(SELECT s."UserName"

      FROM "Staffs" AS s

      WHERE s."Status" = 1

      LIMIT 1), OLD.\*;

            RETURN OLD;

        ELSIF (TG\_OP = 'UPDATE') THEN

            INSERT INTO "ReservationOperationsLogs" SELECT 'U', now(),( SELECT s."UserName"

      FROM "Staffs" AS s

      WHERE s."Status" = 1

      LIMIT 1), NEW.\*;

            RETURN NEW;

        ELSIF (TG\_OP = 'INSERT') THEN

            INSERT INTO "ReservationOperationsLogs" SELECT 'I', now(), (SELECT s."UserName"

      FROM "Staffs" AS s

      WHERE s."Status" = 1

      LIMIT 1), NEW.\*;

            RETURN NEW;

        END IF;

        RETURN NULL;

    END;

$$

LANGUAGE plpgsql;

CREATE TRIGGER reservation\_audit

AFTER INSERT OR UPDATE OR DELETE ON "Reservations"

    FOR EACH ROW EXECUTE PROCEDURE process\_reservation\_operations();

Δημιουργία πίνακα log και trigger για τα operations στον πινακα Staffs:

CREATE OR REPLACE FUNCTION process\_staff\_operations()

RETURNS TRIGGER AS $$

    BEGIN

        IF (TG\_OP = 'DELETE') THEN

            INSERT INTO "StaffOperationsLogs" SELECT 'D', now(),(SELECT s."UserName"

      FROM "Staffs" AS s

      WHERE s."Status" = 1

      LIMIT 1), OLD.\*;

            RETURN OLD;

        ELSIF (TG\_OP = 'UPDATE') THEN

            INSERT INTO "StaffOperationsLogs" SELECT 'U', now(),( SELECT s."UserName"

      FROM "Staffs" AS s

      WHERE s."Status" = 1

      LIMIT 1), NEW.\*;

            RETURN NEW;

        ELSIF (TG\_OP = 'INSERT') THEN

            INSERT INTO "StaffOperationsLogs" SELECT 'I', now(), (SELECT s."UserName"

      FROM "Staffs" AS s

      WHERE s."Status" = 1

      LIMIT 1), NEW.\*;

            RETURN NEW;

        END IF;

        RETURN NULL;

    END;

$$

LANGUAGE plpgsql;

CREATE TRIGGER staff\_audit

AFTER INSERT OR UPDATE OR DELETE ON "Staffs"

    FOR EACH ROW EXECUTE PROCEDURE process\_staff\_operations();

Επιστρέφει τα logs του Customers

CREATE OR REPLACE FUNCTION customerlogs()

    RETURNS TABLE (

        operation TEXT,

        time\_stamp TIMESTAMP,

        userId TEXT,

        customerId BIGINT,

        firstName TEXT,

        lastName TEXT,

        email TEXT,

        phoneNumber BIGINT

)

AS $$

BEGIN

    RETURN QUERY SELECT c."Operation", c."Timestamp", c."UserId",  c."Id", c."FirstName", c."LastName", c."Email", c."PhoneNumber"

      FROM "CustomerOperationsLogs" AS c;

END; $$

LANGUAGE 'plpgsql';

Επιστρέφει τα logs του Reservations

CREATE OR REPLACE FUNCTION reservationlogs()

    RETURNS TABLE (

        operation TEXT,

        time\_stamp TIMESTAMP,

        userId TEXT,

        reservationId BIGINT,

        customerId BIGINT,

        roomId BIGINT,

        checkInDate TIMESTAMP,

        checkOutDate TIMESTAMP,

        totalCost DECIMAL

)

AS $$

BEGIN

    RETURN QUERY SELECT c."Operation", c."Timestamp", c."UserId",  c."Id", c."CustomerId", c."RoomId", c."CheckInDate", c."CheckOutDate",c."TotalCost"

      FROM "ReservationOperationsLogs" AS c;

END; $$

LANGUAGE 'plpgsql';

Επιστρέφει τα logs του Rooms

CREATE OR REPLACE FUNCTION roomlogs()

    RETURNS TABLE (

        operation TEXT,

        time\_stamp TIMESTAMP,

        userId TEXT,

        roomId BIGINT,

        floorNumber INT,

        roomNumber INT,

        roomTypeId BIGINT

)

AS $$

BEGIN

    RETURN QUERY SELECT c."Operation", c."Timestamp", c."UserId",  c."Id", c."Floor", c."RoomNumber", c."RoomTypeId"

      FROM "RoomOperationsLogs" AS c;

END; $$

LANGUAGE 'plpgsql';

Επιστρέφει τα logs του Customers

CREATE OR REPLACE FUNCTION stafflogs()

    RETURNS TABLE (

        operation TEXT,

        time\_stamp TIMESTAMP,

        userId TEXT,

        staffId BIGINT,

        firstName TEXT,

        lastName TEXT,

        userName TEXT,

        email TEXT,

        password TEXT,

        phoneNumber BIGINT,

        roleId BIGINT,

        status INT

)

AS $$

BEGIN

    RETURN QUERY SELECT c."Operation", c."Timestamp", c."UserId",  c."Id", c."FirstName", c."LastName", c."UserName",c."Email",c."Password",c."PhoneNumber",c."RoleId",c."Status"

      FROM "StaffOperationsLogs" AS c;

END; $$

LANGUAGE 'plpgsql';