

SISTEMAS DE INFORMAÇÃO E BASES DE DADOS



Assignment 3

Using the Database

Authors:

Niewiński Łukasz 95018

Lewandowska Gabriela 94898

Gualdi Philipp 95059

Group:

47

Lisboa, December 6, 2019

1 Introduction

The developed code during the project can be found under the link:

https://github.com/philippgualdi/sibd_project_3.git

The project is deployed on the Fenix server and can be found under the link:

<http://web.tecnico.ulisboa.pt/ist195018/clients.php>

2 Database

2.1 Database Triggers

Update Client age:

```
CREATE TRIGGER update_client_age
AFTER INSERT ON appointment
FOR EACH ROW
BEGIN
    IF (SELECT YEAR(birth_date)
        FROM client
        WHERE VAT=new.VAT_client) < (SELECT YEAR(NOW())) THEN
        UPDATE client SET age=YEAR(NOW()) - YEAR(birth_date)
        WHERE VAT=new.VAT_client;
    END IF;
END;
```

Ensure individuals has not more positions:

```
CREATE TRIGGER check_new_doctor
BEFORE INSERT ON doctor
FOR EACH ROW
BEGIN
    DECLARE receptionist INT unsigned DEFAULT 0;
    DECLARE nurse INT unsigned DEFAULT 0;

    SELECT COUNT(*) INTO receptionist
    FROM receptionist WHERE VAT = NEW.VAT;
    SELECT COUNT(*) INTO nurse
    FROM nurse WHERE VAT = NEW.VAT;
```

```

IF receptionist > 0 THEN
    SIGNAL SQLSTATE '20004' SET MESSAGE_TEXT =
        'The Employee is already a receptionist';
ELSEIF nurse > 0 THEN
    SIGNAL SQLSTATE '20004' SET MESSAGE_TEXT =
        'The Employee is already a nurse';
END IF;
END;

```

```

CREATE TRIGGER check_new_receptionist
BEFORE INSERT ON receptionist
FOR EACH ROW
BEGIN
    DECLARE doctor INT unsigned DEFAULT 0;
    DECLARE nurse INT unsigned DEFAULT 0;

    SELECT COUNT(*) INTO doctor
    FROM doctor WHERE VAT = NEW.VAT;
    SELECT COUNT(*) INTO nurse
    FROM nurse WHERE VAT = NEW.VAT;

    IF doctor > 0 THEN
        SIGNAL SQLSTATE '20004' SET MESSAGE_TEXT =
            'The Employee is already a doctor';
    ELSEIF nurse > 0 THEN
        SIGNAL SQLSTATE '20004' SET MESSAGE_TEXT =
            'The Employee is already a nurse';
    END IF;
END;

```

```

CREATE TRIGGER check_new_nurse
BEFORE INSERT ON nurse
FOR EACH ROW
BEGIN
    DECLARE doctor INT unsigned DEFAULT 0;
    DECLARE receptionist INT unsigned DEFAULT 0;

    SELECT COUNT(*) INTO doctor
    FROM doctor WHERE VAT = NEW.VAT;
    SELECT COUNT(*) INTO receptionist
    FROM receptionist WHERE VAT = NEW.VAT;

    IF doctor > 0 THEN
        SIGNAL SQLSTATE '20004' SET MESSAGE_TEXT =
            'The Employee is already a doctor';
    ELSEIF receptionist > 0 THEN
        SIGNAL SQLSTATE '20004' SET MESSAGE_TEXT =

```

```

        'The Employee is already a receptionist';
    END IF;
END;

# 2b)
CREATE TRIGGER check_permanent_doctors
BEFORE INSERT ON permanent_doctor
FOR EACH ROW
BEGIN
    IF (SELECT COUNT(*) FROM trainee_doctor
        WHERE VAT = NEW.VAT) > 0 THEN
        SIGNAL SQLSTATE '20001' SET MESSAGE_TEXT =
            'The new Permanent Doctor ID already existst in Trainee Doctor table';
    END IF;
END;

CREATE TRIGGER check_trainee_doctors
BEFORE INSERT ON trainee_doctor
FOR EACH ROW
BEGIN
    IF (SELECT COUNT(*) FROM permanent_doctor
        WHERE VAT = NEW.VAT) > 0 THEN
        SIGNAL SQLSTATE '20002' SET MESSAGE_TEXT =
            'The new Trainee Doctor ID already existst in Permanent Doctor table';
    END IF;
END;

```

All this triggers for checking the employees also exists and get executed if the values in the tables get updated.

Employees and Clients has not the same phone number

```

CREATE TRIGGER check_phonenumber_employee
BEFORE INSERT ON phone_number_employee
FOR EACH ROW
BEGIN
    IF (SELECT COUNT(*) FROM phone_number_client
        WHERE phone = NEW.phone) > 0 THEN
        SIGNAL SQLSTATE '20005' SET MESSAGE_TEXT =
            'A client already uses this phonenumber.';
    END IF;
END;

CREATE TRIGGER check_phonenumber_client

```

```

BEFORE INSERT ON phone_number_client
FOR EACH ROW
BEGIN
    IF (SELECT COUNT(*) FROM phone_number_employee
        WHERE phone = NEW.phone) > 0 THEN
        SIGNAL SQLSTATE '20005' SET MESSAGE_TEXT =
            'An employee already uses this phonenumber.';
    END IF;
END;

```

2.2 Database Functions and Procedures

Function to compute to total number of no-shows

Write a function to compute to total number of no-shows (i.e., appointments where the client missed the consult) for clients of a given gender, within a given age group, and within a given year (i.e., the gender, year, and upper/lower limits for the age should all be provided as parameters).

```

CREATE FUNCTION count_no_shows(
    gender ENUM ('man', 'woman', ''),
    year INT,
    age_lower INT,
    age_upper INT
)
RETURNS INT
NOT DETERMINISTIC
BEGIN
    RETURN (SELECT COUNT(*)
        FROM appointment a LEFT JOIN client c2 on a.VAT_client = c2.VAT
        WHERE YEAR(a.date_timestamp) = year
            AND c2.gender = gender
            AND c2.age BETWEEN age_lower AND age_upper
            AND (a.date_timestamp, a.VAT_doctor) NOT IN (
                SELECT c.date_timestamp, c.VAT_doctor
                FROM consultation c));
END ;

```

Procedure to change a doctors salary

Write a stored procedure to change the salary of all doctors that have been practicing for more than x years, where x is an input parameter. The new salary should correspond to a raise of 10 percent over the original salary, in the case of doctors with more than 100 consults in the current year, and to a raise of 5 percent otherwise.

```

CREATE PROCEDURE change_salary(IN year INT)

```

```

BEGIN

    UPDATE employee SET salary = salary * 1.1
    WHERE VAT IN (SELECT permanent_doctor.VAT
                  FROM permanent_doctor
                  WHERE years > year);

    UPDATE employee SET salary = salary * 1.05
    WHERE VAT IN (SELECT VAT_doctor
                  FROM consultation
                  WHERE YEAR(date_timestamp) = YEAR(NOW())
                  GROUP BY VAT_doctor
                  HAVING COUNT(*) > 100);
END;

```

3 Additional functions and procedures

For the project we wrote some other use-full functions and procedures

```

CREATE PROCEDURE available_doctors(
    date DATE,
    hour TIME
)
BEGIN
    SELECT e.name as name, d.VAT as vat, d.specialization as spec, d.email as email
    FROM doctor d join employee e on d.VAT = e.VAT
    WHERE d.VAT NOT IN (SELECT a.VAT_doctor FROM appointment a
    WHERE DATE(a.date_timestamp) = date and TIME(a.date_timestamp)=hour);
END;

```

```

CREATE PROCEDURE insert_client(
    VAT_          char(15),
    name_         char(30),
    birth_date_   date ,
    street_       char(30),
    city_         char(30),
    zip_          char(15),
    gender_       ENUM ('man', 'woman', ''),
    phone_number_ char(15)
)
BEGIN
    IF NOT EXISTS(SELECT * FROM client c WHERE c.VAT = VAT_) THEN
        START TRANSACTION;
        INSERT INTO client (VAT, name, birth_date, street, city, zip,
            gender, age)

```

```

VALUES (VAT_, name_, birth_date_, street_, city_, zip_,
gender_, YEAR(NOW()) - YEAR(birth_date));
    IF phone_number_ != NULL THEN
        INSERT INTO phone_number_client VALUE(VAT_, phone_number_);
    END IF;
COMMIT;
ELSE
    SIGNAL SQLSTATE '20006'
    SET MESSAGE_TEXT = 'Client already exists in the database.';
END IF;
END;

```

```

CREATE PROCEDURE insert_procedure_in_consultation(
    name_          char(30),
    VAT_doctor_    char(15),
    date_timestamp_ timestamp,
    description_    TEXT
)
BEGIN
    IF NOT EXISTS(SELECT * FROM procedure_in_consultation pic where pic.name = name_
                    and pic.VAT_doctor = VAT_doctor_
                    and pic.date_timestamp = date_timestamp_)
    THEN
        INSERT INTO procedure_in_consultation(name, VAT_doctor,
        date_timestamp, description)
        VALUES (name_, VAT_doctor_, date_timestamp_, description_);
    ELSE
        SIGNAL SQLSTATE '20006'
        SET MESSAGE_TEXT = 'Procedure dental chart for this consultation
        already exists.';
    END IF;
END;

```

```

CREATE PROCEDURE insert_procedure_charting(
    name_          char(30),
    VAT_doctor_    char(15),
    date_timestamp_ timestamp,
    quadrant_      ENUM ('1', '2', '3', '4'),
    number_        int,
    description_    TEXT,
    measure_        TEXT
)
BEGIN
    IF NOT EXISTS(SELECT * FROM procedure_charting pc where
pc.date_timestamp = date_timestamp
                    and pc.name = name_ and pc.VAT = VAT_doctor_

```

```

        and pc.quadrant = quadrant_ and pc.number = number_) THEN
INSERT INTO procedure_charting (name, VAT, date_timestamp, quadrant,
number, description, measure)
VALUES (name_, VAT_doctor_, date_timestamp_,
quadrant_, number_, description_, measure_);
ELSE
    SIGNAL SQLSTATE '20006'
    SET MESSAGE_TEXT = 'Procedure charting already exists in the database.';

END IF;
END;

```

4 PHP backend solutions

Below we present solution for dental chart transaction handling and error handling using PHP and PDO connection.

```

try {
    $connection->beginTransaction();
    // INSERT INTO PROCEDURE IN CONSULTATION
    $message1 = 0;
    $message2 = 0;

    $procedure_in_consultation = $connection->prepare("CALL
insert_procedure_in_consultation('$procedure',
    '$VAT_doctor_p', '$date_p', '$description[0]')");

    if ($procedure_in_consultation->execute()) {
        $result1 = true;
        $message1 = "Procedure in consultation successfully added!";
    } else {
        $result1 = false;
        $message1 = $procedure_in_consultation->errorInfo()[2];
    }
}

$result2 = true;
for ($i = 0; $i < count($_POST['quarter']); $i++) {
    if ($quarter[$i] != "" && $number[$i] != "") {

        // INSERT INTO PROCEDURE CHARTING (INSERT FOR EACH TOOTH)
        $procedure_charting = $connection->prepare("CALL
insert_procedure_charting('$procedure',
    '$VAT_doctor_p', '$date_p', '$quarter[$i]',

```



```

        '$number[$i]', '$desc_tooth[$i]', '$measure[$i]'))");

$message2 = 0;

if ($procedure_charting->execute()) {
    $result2 = true;
    $message2 = "Client successfully added!";
} else {
    $result2 = false;
    $message2 = $procedure_charting->errorInfo()[2];
}

if ($result2==false){
    echo("<script> confirm('$message2');</script>");
    break;
}
}

}
$connection->commit();
$connection = NULL;

echo("<p>Procedure in consultation: $message1</p>");
echo("<p>Procedure charting: $message2</p>");
echo("<hr>");
}
catch (\Exception $e) {
    if ($connection->inTransaction()) {
        $connection->rollback();
        // If we got here our two data updates are not in the database
        echo("<p>Data was not inserted, transaction failed</p>");
    }
}
}

```

5 Webpage

Mainpage

On the main page it is possible to search or add clients and check if a doctor is available for a specific date and time:

Search page for a specific client and add an appointment or see the last Consultations:

In this page for the client a new appointment can be added or old consultations and more information.

If the client shows up on the appointment a new consultation can be added. All appointments in the past without a consultation will be considered as non shown up.

Dental clinic SIBD

New client?

[Add new client](#)

Search for the client:

[Search](#)

Search for available doctors:

[Search](#)

Lisbon, Dec 2019

Figure 1: Main page for search and add clients and doctors

Client search

VAT	Name	Address	Birth date	Treatment history
200000	Holden	San Miguel, 3920 Dui Rd., 99929	1932-05-14	Check history Add appointment

[Add new client](#)

[Back to Clients](#)

Lisbon, Dec 2019

Figure 2: Search page for a specific client.

After a consultation is added the dental chart can be added.

Add consultation

Consultation information

VAT doctor: 100006

Date: 2013-08-15 05:48:27

VAT client: 200057

Assistant nurse:

Choose nurse: Haney ▼

Description:

SOAP-S: enter subjective

SOAP-O: enter objective

SOAP-A: enter assessment

SOAP-P: enter plan

Submit consultation

Figure 3: Webpage add consultation and diagnose

Dental Chart Procedure

Procedure for this consultation does not exists, create new:

You are going to add dental chart procedure for consultation:

VAT client: '400000'

VAT doctor: '100100'

Date and time: '2019-12-20 11:00:00'

Description: describe procedure

Insert procedure charting data - gap per tooth:

Quarter 1	Tooth number 1	Description	Measurement	
Quarter 2	Tooth number 3	Description	Measurement	DELETE
Quarter 2	Tooth number 2	Description	Measurement	DELETE
Quarter 3	Tooth number 1	Description	Measurement	DELETE

ADD TOOTH

SUBMIT FORM

◀ Previous page Back to main page

Figure 4: Webpage add dental chart in a given procedure

Dental Chart Procedure

You are going to add dental chart procedure for consultation:

VAT client: **'400000'**

VAT doctor: **'100100'**

Date and time: **'2019-12-20 11:00:00'**

Procedure name: **'dental_chart'**

Procedure description: **'procedure description'**

Number of teeth added to procedure: 4

quarter = '1', tooth number = '1', tooth desc: "", measurement: "

quarter = '2', tooth number = '3', tooth desc: "", measurement: "

quarter = '2', tooth number = '2', tooth desc: "", measurement: "

quarter = '3', tooth number = '1', tooth desc: "", measurement: "

Validation successful, selected teeth are unique

Procedure in consultation: Procedure in consultation successfully added!

Figure 5: Added dental chart