# CMPE 321
# Summer 2015

Yusuf Hakan Kalayci

Basic Hospital Automation System

**Table Of Content**

## 1. Introduction

I developed a basic hospital automation system with using PHP and MYSQL. It has two types of users : Administrator, Patient. Admin users have ability to edit two different types of area: Branches, Doctors. In other words Admin users are allowed to do some types of operations:

- Listing Branches
- Creating Branch
- Deleting Branch
- Editing Branch
- Listing Doctors
- Creating Doctor
- Deleting Doctor
- Editing Doctor

On the other hand Patient users  are allowed to do some types of types of operations:

- Listing Rendezvous
- Creating Rendezvous
- Deleting Rendezvous
- Editing Rendezvous

## 2. Structure of Tables

| **branch** | |
|---|---|
| id | INT(11) |
| name | VARCHAR(30) |

| **doctor** | |
|---|---|
| id | INT(11) |
| name | VARCHAR(50) |
| birth_date | DATE |
| service_begin_date | DATE |

| **members** | |
|---|---|
| id | INT(11) |
| name | VARCHAR(30) |
| lastname | VARCHAR(30) |

| | |
|---|---|
| email | VARCHAR(50) |
| password | char(128) |
| type | INT(11) |

| **rendezvous** | |
|---|---|
| id | INT(11) |
| time | DATETIME |
| doctor_id | INT(11) |
| patient_id | INT(11) |

| **email** | |
|---|---|
| id | INT(11) |
| time | DATETIME |
| doctor_id | INT(11) |
| patient_id | INT(11) |

## 3. DDL Statements

### 3.1 Foreign Keys

| Name Of Constraint | Table Name | References |
|---|---|---|
| branch_id | doctor | branch |
| doctor_id | rendezvous | doctor |
| patient_id | rendezvous | members |
| doctor_id | email | doctor |
| patient_id | email | members |

### 3.2 Creating Tables

```
CREATE TABLE `members` (
 `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
 `name` VARCHAR(30) NOT NULL,
 `lastname` VARCHAR(30) NOT NULL,
 `email` VARCHAR(50) NOT NULL,
 `password` CHAR(128) NOT NULL,
 `type` INT NOT NULL
);

ALTER TABLE `members` CHANGE `type` `type` INT(11) NOT NULL DEFAULT '1';
```

```
CREATE TABLE `branch` (
 `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
 `name` VARCHAR(30) NOT NULL
);
```

```
CREATE TABLE `doctor` (
 `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
 `name` VARCHAR(50) NOT NULL,
 `branch_id` INT NOT NULL,
 `birth_date` DATE NOT NULL ,
 `service_begin_date` DATE NOT NULL ,
 FOREIGN KEY(branch_id) REFERENCES branch(id)
);
```

```
CREATE TABLE `rendezvous` (
 `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
 `time` TIMESTAMP NOT NULL ,
 `doctor_id` INT NOT NULL,
 `patient_id` INT NOT NULL,
 FOREIGN KEY(doctor_id) REFERENCES doctor(id),
 FOREIGN KEY(patient_id) REFERENCES members(id)
);
```

```
CREATE TABLE `email` (
 `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
 `time` DATETIME NOT NULL ,
 `doctor_id` INT NOT NULL,
 `patient_id` INT NOT NULL,
 FOREIGN KEY(doctor_id) REFERENCES doctor(id),
 FOREIGN KEY(patient_id) REFERENCES members(id)
);
```

**3.3 Saved Procedures**

```
CREATE PROCEDURE PastRendezvous(IN id_val INT)
BEGIN
 SELECT r.id, r.time, d.name AS doctor_name, d.id AS doctor_id
  FROM rendezvous r INNER JOIN doctor d ON r.doctor_id=d.id
    WHERE r.time < now() AND r.patient_id = id_val ORDER BY time ASC;
END
```

```
CREATE PROCEDURE FutureRendezvous(IN id_val INT)
BEGIN
 SELECT r.id, r.time, d.name AS doctor_name, d.id AS doctor_id
  FROM rendezvous r INNER JOIN doctor d ON r.doctor_id=d.id
    WHERE r.time > now() AND r.patient_id = id_val ORDER BY time ASC;
END
```

**3.4 Trigger**

```
CREATE TRIGGER new_rendezvous_added
 AFTER INSERT ON `rendezvous` for each row
  BEGIN
   INSERT INTO email (time,doctor_id,patient_id)
   VALUES (new.time,new.doctor_id,new.patient_id);
  END
```

**4. DML Statements From Source Code**

**i) Branch Operations**

```
//creating branch
$sql = "INSERT INTO branch (name) VALUES ('".$_GET['name']."')";

//deleting branch
$sql = "DELETE FROM branch WHERE id=".$_GET['id'];

//updating branch
$sql = "UPDATE branch SET name='".$_GET['name']."' WHERE id=".$_GET['id'];
```

```
//listing branches
$sql = "SELECT id, name FROM branch";
```

**ii) Doctor Operations**

```
//creating doctor
$sql = "INSERT INTO doctor (name,birth_date,service_begin_date,branch_id)
VALUES ('".$_GET['name']."','" .$_GET['birth_date']. "','".$_GET['service_begin_date']. "','".
$_GET['branch']."')";

//deleting doctor
$sql = "DELETE FROM doctor WHERE id=".$_GET['id'];

//updating doctor
$sql = "UPDATE doctor SET name='".$_GET['name']."', birth_date='". $_GET['birth_date'] ."',
service_begin_date='".$_GET['service_begin_date']."',branch_id='".$_GET['branch']."'WHERE
id=".$_GET['id'];

//listing doctors
$query = "SELECT d.id, d.name, d.birth_date, d.service_begin_date, b.name AS
branch_name, b.id AS branch_id FROM branch b INNER JOIN doctor d ON b.id =
d.branch_id";
```

**iii) Rendezvous Operations**

```
//creating rendezvous
$sql = "INSERT INTO rendezvous (time,doctor_id,patient_id) VALUES ('".$rend_time.
"','".$_GET['doctor_id']."','".$_SESSION['uid']."')";

//deleting rendezvous
$sql = "DELETE FROM rendezvous WHERE id=".$_GET['id'];

//updating rendezvous
$sql = "UPDATE rendezvous SET time='".$rend_time."', doctor_id='". $_GET['doctor_id']
."'WHERE id=".$_GET['id'];

//listing past rendezvous
$query = "CALL PastRendezvous(".$_SESSION['uid'].")";

//listing future rendezvous
```

```
$query = "CALL FutureRendezvous(".$_SESSION['uid'].")";
```

**5. PHP Side**

In addition to SQL side for some properties of sites we developed some functions on PHP side.

First of all, on user login and after login to give ability to users for seeing site information we use Sessions. With using this, users can login to site and until this session expire or they log out they can freely use their profile.

On the other hand, we don't want that users who is not admin see admin pages and vice versa. Then we disallowed this types of operations.

We protect our sites from sql injection attacks with using some types of mysql operations on our main page.

Finally, we created a trigger to Mysql. In addition on rendezvous this trigger runs and adds an new element to email table which stores the information of email which will be sent. Besides, we create an PHP file which sends emails to people who is needed. This PHP file is running as cronjob in every minutes(it can be readjusted). This php file sends emails which is stored in email table and remove them from table.

**6. Conclusion**

This provide give me enormous information about database system, creating database for specified purpose, relations between tables, different types of operations in sql such as triggers, saved procedures. Moreover it is the first time that I use Php , so it also increase the my web based programming language ability.

In the future, this project will help me about much thing, for example I can classify a database is well-constructed or not, coding is efficient or not or what will be possible to do in this project.