CMPUT 391 Project - MedWave | Waves for the Future
Winter 2013 | University of Alberta
Jeremy Smereka
Amir Salimi

Introduction

Within the medwave folder, there are two folders Media and System. The Media folder contains the CSS code, images and the visual aspects of the site. System contains the architecture and functionalities of the site.

Login Module

index.php: The page where the users logs in.
It sends the data to the **controller/user.php**. Where the authenticate() function gets the supplied username/password and checks the users table to see if they match using the following sql command:

```
SELECT COUNT(*) AS count, class, user_name
FROM users
WHERE user_name=:username AND password=:password
```

If the authentication is completed, the user gets redirected to the home page under **home.php**.

User Management Module

This module allows a system administrator to manage (to enter or update) the user information, i.e., the information stored in tables users, persons, family_doctor. The admin can click on manage users on their home page and access this module.

The file **user-list.php** is called by default, it can be accessed by the administrators to modify/add/remove data from tables persons and users (these 2 tables are joined to make adding and removing users easier) and the family_doctor table. Clicking on update next to any row will take all the data out of that row and send it to user.php. where updateUser() or addUser() will do the following:

First update or add all the personal information:

```
UPDATE persons
SET first_name=:fname,
        last_name=:lname,
        address=:address,
        email=:email,
        phone=:phone
WHERE user_name=:name
Or if it's new information

INSERT INTO persons (first_name, last_name, address, email, phone,
user_name)
```

```
VALUES (:fname, :lname, :address, :email, :phone, :name)
```

Then update the users table.

```
UPDATE family_doctor
SET patient_name=:patient
WHERE doctor_name=:doctor
AND patient_name=:oldpatient
```

or

```
INSERT INTO users (user_name, password, class, date_registered)
VALUES (:username, :password, :class, :date_registered)
```

On the family_doctor table, only removing/adding can be done. if the user clicks on add, the date will be send to controller/user.php where the function addDoctor() does the following:

```
INSERT INTO family_doctor (family_doctor, patient_name)
VALUES (:doctor, :patient)
```

Same for removing, but the sql command is:

```
DELETE FROM family_doctor
WHERE patient_name=:patient
AND doctor_name=:doctor
```

Report Generating Module

The report-gen.php is the view where the user may enter the report generating parameters. The data will then be send to _report.table.php where the sql statement is executed and presented.

Report Generating is done by selecting a diagnosis and a date range (from/to). From here the following SQL is executed:

```
SELECT p.user_name AS username,
           p.first_name As nameF,
           p.last_name As nameL,
           p.address As address,
           p.phone AS phone,
           MIN(r.test_date) AS testDate,
           r.diagnosis As diagnosis
FROM persons p
RIGHT JOIN radiology_record r
ON p.user_name = r.patient_name
WHERE r.diagnosis=:diagnosis AND r.prescribing_date BETWEEN :from
AND :to
```

This joins the persons table with a radiology record, which is then used to display the results of the user

information, provided it is available.

Uploading Module

Enables the user input all the information for a patient record and select multiple images to upload. The information will be sent to **controller/upload.php** function. The information will then be uploaded with the following commands:

```
INSERT INTO radiology_record (record_id, patient_name, doctor_name,
radiologist_name, test_type, prescribing_date, test_date, diagnosis,
description)
 VALUES
(:record_id, :patient_name, :doctor_name, :radiologist_name, :test_ty
pe, :prescribing_date, :test_date, :diagnosis, :description)
```

This same record is then also partially inserted into the radiology_search table which is a different database type so to allow for searching.

```
INSERT INTO radiology_search (record_id, patient_name, diagnosis,
description)
VALUES (:record_id, :patient_name, :diagnosis, :description)
```

Finally, the images are stored into the database.

```
INSERT INTO pacs_images (record_id, image_id, thumbnail, regular_size,
full_size)
VALUES (:record_id, :image_id, :thumb, :regular_size, :full_size)
```

Search Module

Enables the user input either text OR a From/To Date range. Once either is submitted, the results being displayed will depend upon user class and their search parameters.

The search-results file gets the search parameters, and sends the data to one of the functions under the _search folder depending on the user class.

The following SQL statements are the primary driving force behind it.

```
SELECT r.record_id AS record_id,
      r.patient_name AS patient_name,
      r.doctor_name AS doctor_name,
      r.radiologist_name AS radiologist_name,
      r.test_type AS test_type,
      r.prescribing_date AS prescribing_date,
      r.test_date AS test_date,
      r.diagnosis AS diagnosis,
      r.description AS description,
      MATCH(s.patient_name) AGAINST(:search IN BOOLEAN MODE) AS freq1,
```

```
      MATCH(s.diagnosis) AGAINST(:search IN BOOLEAN MODE) AS freq2,
      MATCH(s.description) AGAINST(:search IN BOOLEAN MODE) AS freq3
FROM radiology_search s INNER JOIN radiology_record r ON
s.record_id=r.record_id
WHERE MATCH(s.patient_name, s.diagnosis, s.description)
AGAINST(:search IN BOOLEAN MODE)
ORDER BY (6*freq1)+(3*freq2)+(freq3) DESC
```

The above query, is modified minorly for each class of user. We introduce a where clause for each class EXCEPT admin (shown) which filters based on user name (i.e. if class is patient, it goes `WHERE patient_name=:patient`).

Having used MySQL we found that we needed to use a separate table to power it, as InnoDB does not have Full Text search.

ASSUMPTION: We assume Doctors ONLY have records of their patients, thus `WHERE doctor_name=:doctor` is what we used for the Doctors.

Data Analysis Module

The analysis.php enables the user to put in their analysis parameters which will then be sent to _analysis/form.data.php which executes the sql queries.

Data Analysis is a complicated module for us since we used MySQL. For the data analysis, after the user chooses the Patients, Diagnosis, and Time Periods, we then do the following:

```
CREATE TEMPORARY TABLE IF NOT EXISTS olap_analysis
    ENGINE=MEMORY AS (
                SELECT COUNT(i.image_id) AS imgCount,
               r.patient_name AS patient_name,
               r.test_type AS test_type,
               r.test_date AS test_date
        FROM radiology_record r
        INNER JOIN pacs_images i
        ON r.record_id=i.record_id
        GROUP BY i.record_id)
```

This creates a temporary table in memory with the data from the inner select query, which is the count of images, and required information for the OLAP.

From here, we begin the actual processing of the data (subsetting). Since MySQL doesnâ€™t have a cube function this was abit trickier for us. All the queries contain the following:

```
SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis
```

After this point, the queries change based on what is being requested. If a specific patient/testType is requested, then `WHERE` is added on. If ALL patients/testTypes are requested, then a `GROUP BY` is

added on for those columns. If only one time is specified then a `WHERE` is added on with the condition being >= or <= depending on if its from or to. Finally, if both from/to is submitted, then it uses `WHERE test_date BETWEEN from AND to`.

The queries are listed below. This most likely is not the most efficient way of doing this but in our situation it is what we came up with. Note: $spec is the Roll Up/Drill Down operation, where $spec == YEAR || MONTH || WEEK.

```
SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE patient_name=:patient

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis GROUP BY patient_name

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE test_type=:testType

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis GROUP BY test_type

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE test_date>=:fromDate GROUP BY
".$spec."(test_date)

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE test_date<=:toDate GROUP BY
".$spec."(test_date)

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE test_date BETWEEN :fromDate AND :toDate
GROUP BY ".$spec."(test_date)

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE patient_name=:patient AND
test_type=:testType

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE test_type=:testType GROUP BY patient_name

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE patient_name=:patient GROUP BY test_type

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis GROUP BY patient_name, test_type
```

```
SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE patient_name=:patient AND
test_date>=:fromDate GROUP BY ".$spec."(test_date)

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE test_date>=:fromDate GROUP BY patient_name,
".$spec."(test_date)

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE patient_name=:patient AND test_date<=:toDate
GROUP BY ".$spec."(test_date)

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE test_date<=:toDate GROUP BY patient_name,
".$spec."(test_date)

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE test_type=:testType AND test_date>=:fromDate
GROUP BY ".$spec."(test_date)

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE test_type=:testType AND test_date<=:toDate
GROUP BY ".$spec."(test_date)

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE test_date>=:fromDate GROUP BY test_type,
".$spec."(test_date)

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE test_date<=:toDate GROUP BY test_type,
".$spec."(test_date)

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE patient_name=:patient AND
test_type=:testType AND test_date>=:fromDate GROUP BY
".$spec."(test_date)

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE patient_name=:patient AND
test_type=:testType AND test_date<=:toDate GROUP BY
".$spec."(test_date)

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE test_type=:testType AND test_date>=:fromDate
```

```
GROUP BY patient_name, ".$spec."(test_date)

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE test_type=:testType AND test_date<=:toDate
GROUP BY patient_name, ".$spec."(test_date)

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE patient_name=:patient AND
test_date>=:fromDate GROUP BY test_type, ".$spec."(test_date)

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE patient_name=:patient AND test_date<=:toDate
GROUP BY test_type, ".$spec."(test_date)

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE test_date>=:fromDate GROUP BY test_type,
patient_name, ".$spec."(test_date)

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE test_date<=:toDate GROUP BY test_type,
patient_name, ".$spec."(test_date)

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE patient_name=:patient AND test_date
BETWEEN :fromDate AND :toDate GROUP BY ".$spec."(test_date)

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE test_date BETWEEN :fromDate AND :toDate
GROUP BY patient_name, ".$spec."(test_date)

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE test_type=:testType AND test_date
BETWEEN :fromDate AND :toDate GROUP BY ".$spec."(test_date)

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE test_date BETWEEN :fromDate AND :toDate
GROUP BY test_type, ".$spec."(test_date)

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE patient_name=:patient AND
test_type=:testType AND test_date BETWEEN :fromDate AND :toDate GROUP
BY ".$spec."(test_date)

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
```

```
FROM olap_analysis WHERE test_type=:testType AND test_date
BETWEEN :fromDate AND :toDate GROUP BY patient_name,
".$spec."(test_date)

SELECT SUM(imgCount) AS imgCount, patient_name, test_type, test_date
FROM olap_analysis WHERE patient_name=:patient AND test_date
BETWEEN :fromDate AND :toDate GROUP BY test_type, ".$spec."(test_date)
```