CS3101 - P2 Report

190022658

April 12, 2022

1 Usage

The practical is built using MariaDB, JavaScript, NodeJS, Express and HTML.

To run the node server follow the steps below

- set PORT environment variable as your uid
- npm i
- npm run setup
- mysql -h <hostname> -u <username> -p <database-name> < src/database/func.sql
- mysql -h <hostname> -u <username> -p <database-name> < src/database/views.sql
- npm start

NOTE: the sequence is important as views.sql depends on func.sql.

2 Overview

This practical touched upon all aspects of database design and I have completed all of the basic functionality.

- Created a MariaDB database
- Added tables
- Uploaded sample data
- Created Views and a Function
- Enforced all the basic constrains

- Created a procedure
- Designed a backend which connects the database to a front end

3 Database Implementation

3.1 Tables

The only things to be noticed are the use of DELETE CASCADE. The constrains were also set at the same time.

3.2 Views

There was a need for join to form views. I also made a helper sql function called winner. To assist with finding out the number of wins a person has. *view_ contact_ details* uses two system defined functions. I was lucky enough to find GROUP_CONCAT() which combined all the phone number to help me get the desired output.

3.3 Queries

Queries were used in views, function, procedures and also the backend. There was one slightly complicated one where we had to show match details but had to display the names instead of the email. There was two joins necessary to do that as there were two players in the relation. Each of there were given a different identifier to distinguish between them.

3.4 Functions

This was one of the two harder things in this practical along with the procedure. While there might be other better ways to find out the winner using match_id, this is what I came up with. It utilizes declare, count and case to do that.

3.5 Procedures

The procedure was to insert and update data. The difference with functions was clear with the SET keyword. Here, I first declared the variables that I would have to use later. I found LAST_INSERT_ID() after some research which allows me to get the last id that was inserted that I needed as a foreign key for played_set. There were two players but (with the help of winner())

we only kney who won. So using a CASE, I found out the one primary key for player and update it's elo according to the formula given.

4 GUI Implementation

The GUI is HTML combined with JavaScript. It is a very simple layout without styling. There is a main page ('/') which connects to the two basic requirements ('/venue.html' and '/player.html'). The venue page lets you see who and when are playing in specific locations in a simple table. The player page allows you to add players via a form which has very basic validation.