CST363 Introduction to Database Systems

Project 1 – Degree Progress Checker

By Ivan Alejandre
Randy Son

Overview

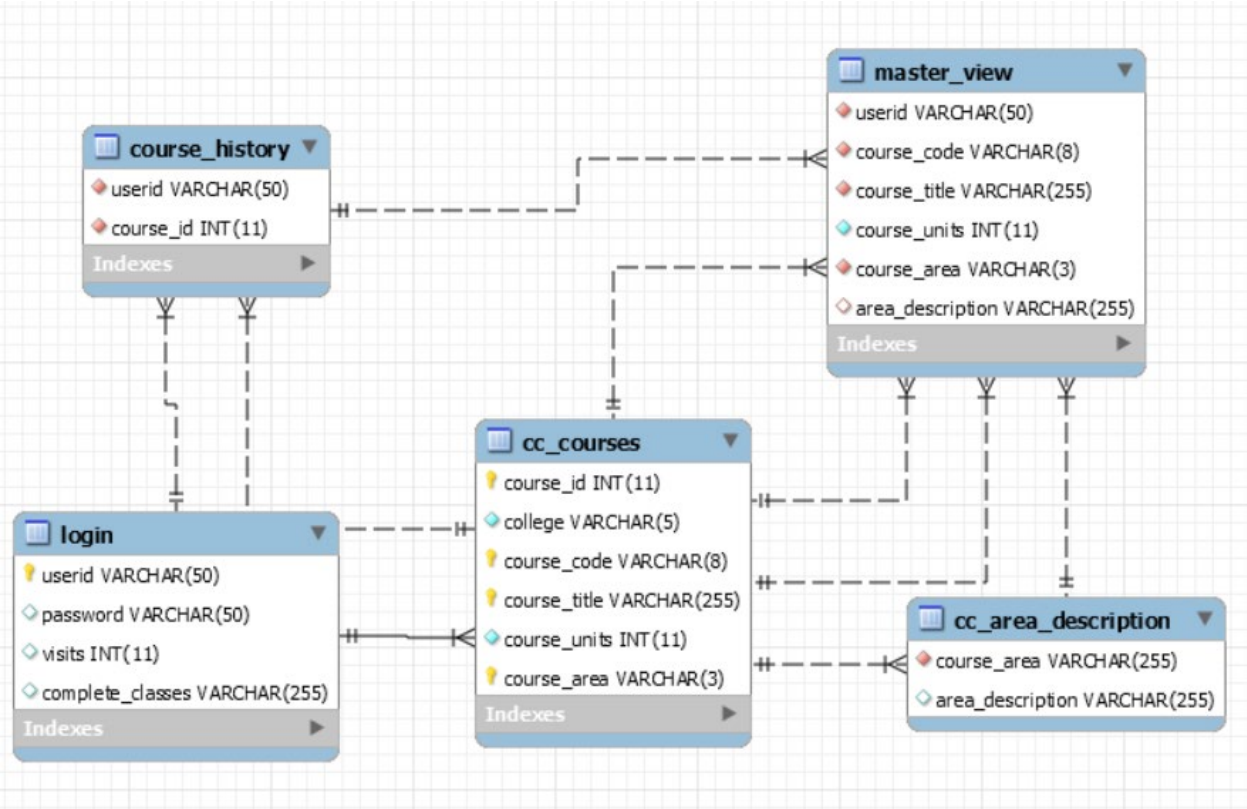
In the current state, we students in the CSUMB Computer Science program utilize a Word document called an Individual Learning Plan or ILP for short to obtain an overview of our degree progress. This file details the classes we have taken and which of them satisfy which areas for general education requirements for Cal State and each one of them is a lengthy and manual process to complete. Multiply that effort with the number of students in the program. Wouldn't it be easier to have an app that automates that for you?

That is where this app comes in. It seeks to reduce the amount of work needed by an academic adviser and provides a proof of concept that a degree roadmap is accomplishable.

To facilitate this, we took class list data from Assist, and parsed it into a tabular format that is ideal for a MySQL database.

In connection with the data, we utilized a similar login system from an earlier assignment that mimics a student signing into the college web portal. The web application, after verifying credentials, then executes a query for each course area, e.g. (A, B, C, D, E) and displays the available classes for each area. Then after having selected the classes they have completed, we store this data in the database for future use. The python script executes conditionals based on what the student selected. Then the student is able to see how many units they have left to satisfy each general education area.

ER Diagram



Currently, the tables in our schema are normalized in that we sought to reduce duplicate data across the schema. The master_view table is created from the following query:

```
CREATE TABLE cst363.master_view
        SELECT
                l.userid
                ,ccc.course_code
                ,ccc.course_title
                ,ccc.course_units
                ,ccc.course_area
                ,ccad.area_description
        FROM cst363.login l
        NATURAL JOIN cst363.course_history ch
        NATURAL JOIN cst363.cc_courses ccc
        NATURAL JOIN cst363.cc_area_description ccad
;
```

Addressing Instructor Feedback from Part 1

We tested the cookie function in Chrome, Firefox, and Chromium, and the cookie issue has not occurred for us. We also talked to our other group mates who used the same function as us (provided by the professor on the forums) and they did not have any errors using cookies this way.

We tested our webpages and scripts by opening several tabs and logging into the database without error. This test ensured that the previous cookies were not deleted.

Operational tables • **at least 3 tables with primary and foreign keys** • **sql script to create table and load with initial data**	**20 points**	
Web application • **2 or 3 (or more) html file and CGI *.py scripts for OLTP**	**20**	
PDF document showing • **Brief explanation of your application, its design and web application** • **ER diagram of operational tables** • **tables are normalized or explained why you chose not to** • **pages are numbers** • **title page with names of team members**	**10**	
Git account containing • **pdf file** • **sql script files** • **html and py files** • ***.mdb model file from Workbench**		
OLAP tables • **star schema design** • ***.mdb model file from Workbench** • **updated PDF with ER diagram of the OLAP tables** • ***.sql scripts to extract and load the OLAP tables**	**40**	
OLAP queries • ***.sql script with 5 select queries that might be typical of the kind of question that an OLAP user might ask. Use comments in the script to explain what the query is asking.**	**40**	