CSI 360 - Database Systems

In this project, we are going to create a database and corresponding website for a course-student database. We will need create three new tables. Those are described below. In addition, we will create a suite webpages capable of inserting data into any one of the tables. In the end, we will also build a webpage to retrieve information from a joined version of this database. Requirements for tables and information is listed below. The tables, along with their attributes and functional dependencies are listed below. In addition, the php and webpage files that you must create are listed below. Finally, this project description concludes with a breakdown of points for this homework assignment.

1. Tables
   (a) Students
      i. Attributes: ID, First Name, Last Name
      ii. FDs: ID $\rightarrow$ First Name, Last Name
      iii. Sample Student: [1, 'Doctor', 'Ramsey']
   (b) Classes
      i. Attributes: ID, Name, Dept, Number, Section Number
      ii. FDs: Dept, Number $\rightarrow$ Name
      ID $\rightarrow$ Dept, Number, Section Number
      iii. Sample Class: [1, 'Databases','CSI',360,10]
   (c) Attending (Junction Table for Students Attending a Class)
      i. Attributes: ID (from Students table), ID (from Classes table)
      ii. Sample Attending: [1, 1]

2. Websites

   (a) make.php - drops all tables, creates all tables, adds at least two sample rows to each table. This page should also clearly describe/define who is working on this project. This page is also your summary page. List all members of your team. Clearly describe the following in this page:

      i. Status of your project - be thorough
      ii. Most challenging aspect of your project
      iii. Any deviations from the project description (BCNF for example)

   (b) data.php - this webpage will have three different forms in it.

      i. The first form will have editable entries for the Students table attributes (except for ID which should be generated automatically by your database). When the submit button for this form is clicked, it will load insert1.php with post data.
      ii. The second form will have editable entries for the Classes table attributes (except for ID which should be generated automatically by your database). When the submit button for this form is clicked, it will load insert2.php with post data.
      iii. The last form will have two editable entries for the Attending table. When the submit for this form is clicked it will load insert3.php with post data.

   (c) insert1.php - post data received from data.php will be used to generate a query to insert a new row into the Students table.

   (d) insert2.php - post data received from data.php will be used to generate a query to insert a new row into the Classes table.

   (e) insert3.php - post data received from data.php will be used to generate a query to insert a new row into the Attending table.

   (f) displayall.php - retrieves all rows from all tables and displays them in a neat fashion.

   (g) list.php - retrieve a course id and send it to join.php via a post. This should be done via a form.

   (h) join.php - given the post data from list.php, display the list of students in the chosen class id. This requires a special query called a join. This can allow you to lump data together from two tables. If table1 has name and table2 has name we might run: SELECT * FROM table1 INNER JOIN table2 ON table1.name=table2.name; This would give all information from table1 and table2 with one column representing the name. In a SELECT you can also request only the information that you like. So, if you'd only like some of the information you can request it by column name. For example: SELECT table1.columnname, table2.columnname INNER JOIN table2 on table1.name=table2.name

3. Points breakdown
   (a) make.php (25 points)
   (b) data.php plus insert1,2,3 (25 points)
   (c) displayall.php (25 points)
   (d) list/join.php (25 points)
4. Failure to do the following per page will result in a loss of points.
   (a) Completes the task requested for that page.
   (b) Displays the information in a clear manner.
   (c) Correctly Escapes Inputs (for inserts/select/join queries)
5. Useful order to work on these pages:
   (a) make.php - get all the tables made with some data in them First
   (b) displayall.php - see if you can get all the data displayed now
   (c) list/join.php - since this doesn't affect the tables, see if you can get this form and query pair to work correctly and display something Useful
   (d) data/insert1.php - Get insert into one table working correctly first
   (e) data/insert2.php - Insert into another table - if you do this in the proper order you can check using the mechanisms above!
   (f) data/insert3.php - Insert into the last table. Check using the stuff above!