I308 Team Final Project, Part Two

For the final team project you will design and build a database for the Student Records System. You have been given some preliminary information about this, but be aware that some of the requirements below are slightly different and that there is additional material.

Although (of course*), this system will be nothing like the size of IU, you can use your experiences here and resources available on the web to help you clarify and interpret the requirements. We will also issue clarifications as necessary as you work on the project.
**no pun intended*

About Part Two

For Part Two of the project, based upon your ERD, you will be creating the database objects, inserting data, and selecting the data to confirm your design in MySQL.

When is it due?

You will need to demo your completed final project to your AIs during your Week 15 Lab and the final submission is due one hour after the end of that lab. During lab, the graders will assess your work, and you are free to make corrections and improvements until one hour after your lab has ended. No submissions will be accepted after that time.

Additionally, a Team Final Project Part Two Progress Report will be due at the beginning of lecture during your Week 13 Lecture. It must be printed and fully filled-out. This will be worth 20 points.

Final Submission Requirements

Submit all files to the Canvas assignment.

- A `teamXX.txt` file containing the URL link to where the PHP/HTML is located. This can be in an individual team member's account.
- A `teamXX.sql` file containing text versions of all the SELECT statements. The final Select statements will be presented to the AIs in class on the due date through the PHP scripts that you create and further analyzed when grading.
- A teamXX.sql file containing text versions of all the CREATE statements.
- A `teamXX.zip` file containing all of your PHP/HTML files.
- A `teamXX.pdf` that contains your teams final ERD (revised for any changes) and final Conversion (also revised for any changes).

When submitting the work, only put the names of team members who contributed to the work. If a member does not do the work they are assigned, the team is allowed to give that member a percentage of the team's final grade. Please note these issues clearly in the submitted document.

Requirements

There is a list of requirements for Part Two of the project.

Progress Report (20 points)

The Team Final Project Part Two Progress Report (“TFPPR”) is located on Canvas. This is due before the full submissions – see the previous page. This needs to be printed off and filled out.

Data Requirements (20 points)

We are not trying to create a whole university the size of IU! Rather, you need to create sufficient data to demonstrate that your design works and that your queries return the appropriate data. You will need to create at least the amount of data shown below.

Once you have decided which queries you plan to write, you will need to ensure that your tables contain data that is capable of demonstrating that your queries work correctly. Do not simply insert multiple identical rows—for example, the same group of students taking the same courses multiple times. This will result in a zero grade for this part of the project. The data should be believable.

Include any other data that is necessary for your system to function. Remember you are creating is to sell the system to a client so it should look as realistic as possible.

Item	Minimum
Students	30
Faculty	10
Majors*	4
Semesters	4
Buildings	3
Classrooms	8
Offices	10
Dorm rooms	10
Departments	3
Courses	20
Sections per Semester**	12
Students taking Classes***	Lots!

** Set graduation requirements low, e.g., 15-20 hours so that some of the students will have graduated.*
*** Include examples of multiple sections of the same course.*
**** This will be the big one! You may need as many as several hundred rows. You can have some small or even empty sections. Be sure to include cases of students following/violating prerequisites and anything else you need to demonstrate your system works.*

Required Queries (50 points total)

Teams will select 50 points worth of queries from the following list with the following restrictions:

1. You must include at least one “b” and one “c” query.
2. You may not select more than one query with the same number.
3. You must choose at least two queries where the results are based on user selection of a value (using dropdowns in PHP). These values are marked with asterisks in the query description.

The options for queries are located at the end of this document. The point value of queries is based on the letter. A queries are worth 5 points, B queries are worth 10 points, and C queries are worth 15 points.

Note: If any of the statements return a SQL error for bad coding, the group will receive a 0 (zero) for this part of the project. Please run/test your scripts before submitting to Canvas!

Additional Queries (10 points)

Along with these select statements, your group needs to provide two or more additional statements. If your database contains tables that have not been used in the above statements, provide select statements (for each table) demonstrating the function of these tables. These statements may not be a the same as the provided questions; instead they should show something unique about your system that the other statements did not demonstrate

PHP (30 points)

In addition to the above requirements, the queries statements must be created in PHP and formatted appropriately to select (using dropdowns) and view the results. This is for the presentation of the system to the client (the AIs) during class.

Revised ERD/Conversion (20 points)

You will turn in a revised ERD and Conversion that represents the final version of your database.

Miscellaneous Things

- MySQL tables for the system will be created in your team database account. Please make sure your table names do not conflict with existing tables that we use in class. If you like, you can prefix your tables with tp_ (for team project) or some other prefix of your choosing. (e.g. `classes` becomes `tp_classes`)

Appendix: Available Queries

Sections

1a Produce a roster for a *specified section* sorted by student’s last name, first name

1b Produce a class roster for a *specified section* sorted by student’s last name, first name. At the end, include the average grade (GPA for the class.)

Classrooms

2a Produce a list of classrooms that are equipped with *some feature*—e.g., “wired instructor station”.

2b Produce a list of classrooms that are equipped with *some feature*—e.g., “wired instructor station”—that are available at a particular time.

Faculty

3a Produce a list of all faculty and all the courses they have ever taught. Show how many times they have taught each course.

3b Produce a list of faculty who have never taught a *specified course*.

Student Registration

4a Produce a list of students who are eligible to register for a *specified course* that has a prerequisite.

4b Produce a list of students who are eligible to register for a *specified course* that has a prerequisite, but do not include those students who have already taken the course that has the prerequisite.

4c Produce a list of all students who took a course that had a prerequisite but the student had not taken the prerequisite. Include the semester, the course subject and number, and the grade the student received.

Student Transcript

5a Produce a chronological list (transcript-like) of all courses taken by a *specified student*. Show grades earned.

5b Produce a chronological list of all courses taken by a *specified student*. Show grades earned. Include overall hours taken and GPA at the end.

5c Produce a chronological list of all courses taken by a *specified student*. Show grades earned. Include overall hours *earned* and GPA at the end. (Hint: An F does not earn hours.)

Appendix: Available Queries

Building Occupancy

6b Produce a list of students and faculty who were in a *particular building* at a *particular time*.
(10 points)

6c Produce a list of students and faculty who were in a *particular building* at a *particular time*. Also include in the list faculty and advisors who have offices in that building.
(15 points)

Dorm Rooms

7a Produce a roster of students who have a room in a *specific dorm”. Include room type. (5 points)

7b Produce a list of dorm room ‘anomalies’—single rooms occupied by more than one student and double rooms occupied by a single student. (10 points)

7c Produce a list of students and amount owed or amount overpaid for students who have paid too much or not fully paid for their dorm room. (Remember that students only pay half the rent for a double room.) (15 points)

Attendance Verification

8b Produce an alphabetical list of students who have not attended during the two most recent semesters along with their parents’ phone number.
(10 points)

Majors

9a Produce a list of majors offered, along with the department that offers them and their requirements to graduate (hours earned and overall GPA).
(5 points)

9b Produce a list of students with hours earned who have met graduation requirements for a *specified major*.
(10 points)

9c Produce a list of students with hours earned and overall GPA who have met the graduation requirements for any major. Sort by major and then by student.
(15 points)