Phase II Grading Rubric & list of common problems

Here are where most teams lost points on Phase II of the project. By reading this set of problem areas and comparing to your project deliverable, you should be able to identify the exact problems you had, but if you have any questions, please visit my office hours. In my opinion, -1 errors are only half as serious as -2 errors.

Relational Schema (diagram) and Create Table Statements:

-1 Redundant Attributes - Having an attribute in two places where it was only needed in one, or making a derived attribute explicit. For example, if your "tutor" is a "Student" and student has a phone number attribute, your tutor does not also need a phone number attribute. If you can determine that a tutor time slot is "hired" by looking to see if it participates in the "hires" relationship, you do not need an explicit "isHired" attribute.

-1 Inconsistencies in Domains - The domain of an attribute does not match that of another attribute that refers to it (foreign key) or should be able to hold the same thing. For example, if your GTID is a VARCHAR 30 in one place and VARCHAR 9 in another.

-1 Extra Attributes in Key - Your key can have one or more attribute removed and still function as a key. For example, the "Hires" relationship can be uniquely identified using TutorID, Semester, Day and Time. (A single tutor can not be available twice at the same semester/day/time.)

-2 Missing Attributes - Your relational schema or tables created by the CREATE TABLE command is unable to represent a piece of information in the EER or project description because you forgot an attribute. (For example, GTID is important in the Student relation.)

-2 Missing Part of Key - Your key is incomplete, and needs at least one more attribute to be an actual key that uniquely identifies every tuple. For example, a tutor can be rated by the same student in the same course only once *PER SEMESTER*. So TutorID, StudentID, Course-School/Number is not sufficient, you also need Semester as part of your key.

-2 Missing Foreign Key Reference - Your schema diagram lists something as a foreign key, and your Create Table statements did not, or visa versa, or it was missing in both.

-2 Inconsistencies Between Diagram & Create Table statements. The left hand didn't know what the right hand was doing. Errors and inconsistencies in documentation and actual implementation causes many costly mistakes.

-2 Failure to implement a relationship - Your Schema/Tables are unable to represent a relationship captured by the project description and/or EER diagram. For example, if a tutor has attributes for the School and Class Number that they tutor, it means that a tutor can only tutor a single course, while the "tutors" relationship is supposed to be 1:N.

SQL Tasks:

Login

-1 Did not save the users GTID into an (explicitly) global variable for other tasks to refer to.

Search for Tutor

-1 Your GUI needs to provide the School/Class# from the DB.
-1 Your output needs to be ordered by tutor ratings
-1 Your task failed to save the tutors with matching timeslots so that the "Schedule a Tutor" task could get them.
-2 Missing Loops: You must check every tutor who is qualified to offer the course, and you must check every day/time that the student is available. (GUI gives you more than a single time/date). SQL either needs to include "IN" or be ran multiple times if using straight equality checks.)

Schedule a Tutor

-1 Must make sure that the user has only chosen a single tutor/timeslot
-2 You must make sure that the student does not already have a tutor scheduled for this school/class#/Semester. (Avoid double dipping).
-2 General SQL errors....A single tutor still has multiple timeslots, you only book one of them!

Student Rate Tutor

-1 - The form must be completely filled out, or you must produce an error.

-1 - The school/course# drop-down should be loaded from the database. You can generate a list of all school/course#'s, but customizing one to only the courses that the particular users has had tutoring from is very slick.
-1 - Errors with abstract code (ordering issues, logic is wrong, etc)

-2 If you did NOT generate a list that ONLY includes courses the student has received tutoring in, you MUST check that they actually used this tutor in the current semester. Also, you should handle what happens if they "re-rate" a tutor. Either update their old rating, or refuse to insert a duplicate rating!

Apply to Tutor

-1 - You should check the basic info (email, name, phone, etc..) and update the database if anything changes (or add it if your administrator has not already.)
-1 - If any information on the form is missing, you must throw an error.
-2 - You must insert the courses that the tutor wants to tutor. (more than one)
-2 - You must insert the timeslots they are available (more than one)

Tutor View Schedule

-1 - You must check that the GTID entered is correct (a tutor, the current user)

-1 - Output must be ordered by weekday / time.

-2 -You must get the tutor name to display on the GUI (lookup using GTID)

-2 When reporting (name, email) information you do it FOR THE STUDENT WHO IS RECEIVING TUTORING, NOT THE TUTOR! [SQL errors...]

Professor Recommendation

-1 - You must check to make sure that all fields are filled in, otherwise display an error

-1 - You should confirm that the GTID entered is actually a tutor in the system.

-2 - If this professor has already rated this tutor, you should either a) refuse to update it, or b) Update it and let the professor know you updated their old rating. (You should NOT enter a duplicate rating).

Administrator Report 1

-1 - If you check each semester with a separate SQL statement, it messes up the ORDER BY. Instead, you should do something along the lines of "WHERE semester = FALL OR SEMSTER =Spring"
-2 - Order by course and then semester.
-2 - Must get the grand total for all courses
-2 Must get the "multi-semester" total on a per-course basis.

Administrator Report 2

-1 - If you check each semester with a separate SQL statement, it messes up the ORDER BY. Instead, you should do something along the lines of "WHERE semester = FALL OR SEMSTER =Spring"
-2 - Did not get the average over all semesters of a particular course.
-2 - Did not get the average on a by-semester basis.
-6 - Did no ordering by course or semester.