Project Phase 2 Deliverables: Summer 2018

Project Phase 2 is a continuation of Project Phase 1. As a result, the documentation for the project does not change, only the deliverables. We occasionally make minor clarifications to the project specification, so it's probably not a bad idea to check it over, although there shouldn't be any significant changes, merely clarifications.

In this phase you will focus on the Design portion of the project. The lectures called "Methodology III: Design" featuring the GTOnline project should give you a fair indication of what to do for this phase. You need to submit the following:

- A brief explanation of the changes made to your Phase 1 submission, if any.
- Updated EER and the EER to Relational mapping correctness. **(25%)**
  <u>Even if you didn't make changes to your EER, you must submit it with Phase 2 or there will be a grading penalty.</u>
  - o Make sure the names you choose for your relations and columns closely match your EER labels so we can easily validate your mapping.
  - o If you don't follow the relational schema examples from the lectures or the sample submission, your grade may be penalized.
- SQL Create Table statements to create your schema (tables with constraints and keys). **(25%)**
- Abstract Code (from Phase 1 with TD removed) with in-line SQL queries added. **(50%)**

Essentially, anything Leo does in the lectures on Design for the GTOnline project you should do for your analysis and design of this project. (Many of you will probably have noticed that the first bullet point does not have a percentage associated with it. That first bullet is important because it will make grading much easier by the TAs. So while it does not explicitly have a point value, if you make the job tougher for your grader, you may end up losing points because of it.)

IMPORTANT NOTE: We do not want a dump file of your SQL (either the queries or the create schema statements), we want the handmade script you used to create it. Please ensure that what you submit is your original SQL script, not something created with a GUI. We expect your Phase 2 and Phase 3 submissions to use a fully featured application-independent *relational* DBMS - MySQL, PostgreSQL, Oracle, MS SQL, etc. Something that uses SQL as a query language but is not attached to an RDBMS is not allowed.
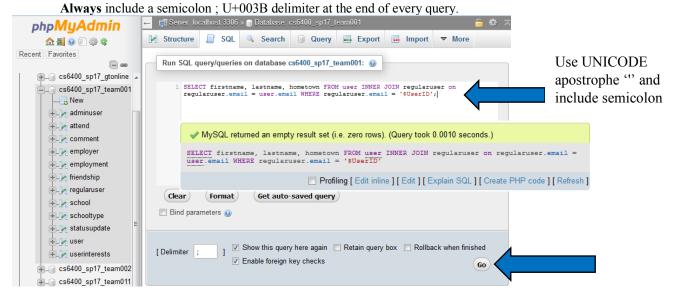
All inline SQL queries should be written to retrieve the exact set of results from the database. **"SELECT *" is bad practice since you are relying on application logic to parse out the result set and not writing targeted queries, which is what we want you to learn.** For any reports that you create, write a **single** query that outputs the report whenever possible. Do not run separate queries and then join the results in code – you should let the database do all the heavy lifting of data. We may also test your abstract code SQL to see that it runs successfully against your SQL schema. **You are not required to create indices in this phase.**

Suggested SQL Naming Conventions – you don't have to follow these, but they are helpful to know.

Entity/Table Names (singular noun): UpperCamelCase   Example: RegularUser (note: without the 's' singular)
Attribute Names: lowercase_underscore   Example: start_date
Primary Surrogate Keys (phase3 only): tableNameID   Example: userID (where 'ID' is capitalized)
Foreign Keys: FK_ChildTable_childColumn_ParentTable_parentColumn   Example: fk_AdminUser_email_User_email

Unicode Character Use Warning: watch the use of the bad: U+2018 left/right single quotes '' vs. apostrophe
    Good: Grave Accent (backtick) ` = U+60 (escape character) Apostrophe **'** = U+27 (good for param)
    Bad: Left Single Quote ' = U+2018 or Right Single Quote ' = U+2019 (*queries will fail*)
    Always include a semicolon ; U+003B delimiter at the end of every query.

Use UNICODE apostrophe '' and include semicolon

What To Submit:
As before, you are welcome to put these all in one file or in **four separate files**. All non-SQL files submitted should be in **pdf format**. If you choose to use one file, please name it "team##_p2_report.pdf". If you submit individual files, please use these names:

- team###_p2_**ac+SQL**.pdf
- team###_p2_**eer2rel**.pdf
- team###_p2_**schema**.sql *(only non-pdf file: which may be imported into phpMyAdmin/pgAdmin)*
- team###_p2_**updatedEER**.pdf

(Replace ### with the number of your team, so if you are Team001, files would be "*team001_p2_ac+SQL.pdf*"…)
Please review the sample submission that is posted in Files on Canvas. Your deliverables should be similar to those in the sample submission. **There may be a grade deduction for any files not named correctly or if we cannot clearly determine which work should be graded in your repository.**

How To Submit
1. Create a directory called "Phase 2" in the team GitHub repository we created for your team (do this by committing any file in the directory called "Phase 2" to GitHub).
2. As you work on phase 2, continue to commit and push the files to GaTech's GitHub enterprise server.
3. If you use branching, make sure to move all work onto your master branch in GitHub. If you provide a commit that is not from the master branch, there may be a penalty on your grade.
4. Find the commit ID of the final commit (you are encouraged to do all of your file sharing through GitHub) using any git tool you choose (a simple "git log" will do it). The commit ID is a SHA hash such as: "40da96c63a0a82d71e531cf137b6c95e282f9289"
5. One team member should submit your team number, followed by a comma, and then the commit ID in the submission text for the Project Phase 1 assignment in Canvas.
   So, if I was on Team001 and we were submitting commit *0b0134604ced2182b8b11d7b299ac59dc70ed722*, someone would submit this:
   001, 0b0134604ced2182b8b11d7b299ac59dc70ed722
   (Please submit a full commit ID. Shortened commit ID submissions may receive a penalty. You do not need to put the text into quotes, either.)

Note: any team member will be able to submit the commit ID to be graded. Please make sure every team member agrees on the correct commit ID before submitting it for grading. New work committed after the deadline **will not be graded**. If a team member submits a new commit ID, and other team members request a different commit to be graded, the entire team will face a deduction and we will randomly select one of the submitted commit IDs to grade.

How You Should Work:
To repeat from last time: The point of this course is to learn all phases of relational database development. As a result, we strongly prefer that teams work collaboratively, not dividedly. By which we mean you should not say "OK, I'll take the SQL schema, you go write the abstract code, Steve can update the EER and Shirley can design the relational schema." Rather, you should each work on all parts of the project, all reviewing, all having input, all fully understanding each part of the design.

Regrade Requests: For any project regrade requests, please offer specific evidence as to why points should be returned. Because the grade is "lower than we expected" is not sufficient reason for honoring a grade change. Be sure to include "p# *grade change request*" in the title of your Piazza post and itemized list, so the instructional team do not confuse your regrade request with a question seeking clarification only. **Regrade requests will be accepted for 1 week after grades and feedback are returned,** so be sure to review your grading feedback immediately. If teams make a regrade request after the regrade deadline, it will not be reviewed.

Example Submission Files:
Note: "DUMP_cs6400_fa17_team001.sql" is an example of what NOT to submit for your database schema. Any SQL schema submission which was auto-generated via phpMyAdmin/Workbench export will receive *NO CREDIT* for that portion of the submission **(-25% p2 grade).**

Feel free to post a private question: (Team### + Instructors) if you need further clarification for your team project.

Thank you in advance!

CS6400 Instructional Team