601.315 Databases, Spring 2022
Project Phase E: Project Wrap-Up

Due: Tue, 10 May at 11pm. Use of late days *is NOT* permitted.
Absolutely no late submissions permitted.
Plan to submit your work only a single time on Gradescope.

Phase E Requirements. In this phase, you will turn in your completed course project. This includes full setup scripts and data files for your MySQL database, as well as an html-based launch page which allows a user to interact with your database. In a **required** final project interview with course staff, you will have the opportunity to highlight your work and discuss it.

You've spent time in earlier phases of the project creating and refining the relations and constraints and individual queries in your database design. In this final phase, think carefully about where your database can make use of mechanisms such as PL/SQL stored procedures and functions, error handling, triggers, cursors, etc., if it is not doing so already. You are required to make meaningful use of these mechanisms at least two different places in your final project submission.

Once on the main html page for your project, the user experience must be self-explanatory; you should assume a user landing at your main page has no SQL experience and no specific knowledge of your database design. The user should be presented with a menu of no fewer than 10 options allowing them to query your database to learn answers to questions you posted in Phase A and refined in SQL in Phase D. You must utilize PHP/mysqli to connect and interact with the database, and all query and database update statements must be executed via the **prepared statement** mechanism. (See demo code from class.) Results should be presented in a clear, user-friendly manner with meaningful labels. Use appropriate CanvasJS charts to display your results in a visual manner where possible. You are free to create separate html pages which allow the user to run different groups of queries, but make sure all of this functionality is clearly accessible from the main html page. Your aim should be to incorporate all 15 queries from Part 1 of Phase D; we give a lower bound of 10 menu options for the user, since some queries can be naturally combined together in one visualization.

As you create the user interface for your database, carefully consider the user experience. Your system should ensure that invalid user inputs are caught and reported to the user, rather than allowing malformed queries to execute and potentially harm the database.

Insertion and deletion queries of the kind you wrote in Part 2 of Phase D must also be incorporated as user options available from the main html page from your project. The user should be able to insert and delete tuples which meaningfully modify the contents of your database, so that subsequent answers to at least some of your questions above will be impacted by these insertions and deletions. (This will help the course staff test your functionality.) Perform at least minimal error handling during insertion to protect the integrity of your data and take care to consider the impact of foreign key constraints on the modifications made by the user. A rule of thumb here is that users shouldn't ever see your code crash, but should instead be presented with a meaningful error message when trying to modify the database in a way that violates the constraints of your database.

For a minority of projects, data visualization via CanvasJS is not a natural fit for the presentation of results. If your group does not employ visualization because it does not work naturally for your topic, you must instead incorporate permission-based database access for your database. You must design and implement a mechanism in which you simulate different user levels via input in html, allowing some users read-only access while others have database modification privileges and can perform inserts and deletes. (Implementing permission-based database access is not a requirement for groups who meaningfully employ data visualization.)

IMPORTANT NOTE: Each course staff member (instructor or course assistant) who reviews your submitted work will be using their own accounts on ugrad and dbase to connect to a **copy of your database** created from the scripts and files you submit. They will not access (or grade) any materials that you create in your ugrad or dbase account. This allows the staff to have a complete snapshot of your project at the time of submission for archival purposes, and ensures that staff grade only work completed before the project deadline.

Course staff will take the following steps when setting up their copy of your database:

1. Copy your submission zip file from Gradescope into their own ugrad.cs.jhu.edu account, and unzip the file
2. Identify an empty (but already created) database on dbase.cs.jhu.edu into which they'll load your material by typing one of the two commands from the ugrad account:

   ```
   mysql -h dbase.cs.jhu.edu -u StaffUsername -D DBName -p < setup.sql
   ```

   for installation of the full version and,

   ```
   mysql -h dbase.cs.jhu.edu -u StaffUsername -D DBName -p < setup-small.sql
   ```

   for the small version built for testing purposes.
3. Copy the entire contents of your submitted subfolder `public_html` into their own `public_html` subfolder on ugrad, setting permissions appropriately so that your submitted html files (only) are world readable (meaning `-rw----r--` in ugrad permissions language).
4. Open your main html file JHED1_JHED2.html in a web browser, and executing the menu of your queries you present from that starting point.

You should assume that the staff member reviewing your work has present in their public_html directory on ugrad a file named precisely *conf.php* which contains lines with the following format (but with specific staff usernames, passwords, and database names filled in):

```
<?php
    $dbhost = "dbase.cs.jhu.edu";       // database server name
    $dbuser = "staff_username";         // staff dbase username
    $dbpass = "staff_password";         // staff dbase password
    $dbname = "staff_selected_db_name"; // name of db in staff account
?>
```

Therefore, any PHP code that will connect to the staff member's copy of your database on dbase can use the line `include 'conf.php'` to make the specified PHP variables above available for use during the creation of a MySQL connection to dbase.

If you have concerns or questions during this final phase of the project, contact your assigned course assistant mentor, or the instructor.

Phase E Deliverables. Via Gradescope team submission by the deadline listed above for Phase E, hand in a single .zip file named **JHED1_JHED2.zip**, where JHED1 and JHED2 represent the JHED IDs for the two partners in your group. One partner will submit the work as a team submission upload at the Project Phase E Gradescope link, and will indicate all partner names. Therefore, only one partner should submit. When your .zip file is unzipped, the working directory must contain the following subfolders and files:

- **README**: The top of your `README` plain text file must list the names and JHED IDs of each partner. Additionally, if course staff (your assigned CA mentor or the instructor) asked to address problematic issues after your Phase C or Phase D submission, write a few sentences to explicitly describe what modifications you made. If no such changes were made, simply note that as your response for this part. Place your response in a file named `README`. Feel free to include in this file any other information you think your CA mentor or the instructor should know.

- a subfolder named **dbase_setup**, which includes:

  - **install-notes.txt**, which contains group members names and JHEDs, and any special instructions necessary for database installation on dbase from ugrad beyond simply running the `mysql` commands described in Step 2 above. For example, if your stored procedures and functions are described in a separate script, describe the appropriate commands to install them, and the order in which they should be executed. If no special instructions are needed, simply include your member names and JHEDs and the text "No special instructions needed." Additionally, if you implemented a form of user access control rather than visualization, please describe your mechanism here, and give instructions regarding which "users" have modify access and which have read-only access to the database.
  - **setup.sql**, an up-to-date version of the SQL setup script needed to create the relations in your database, fully populate them, and create any stored procedures and functions, triggers, etc. to set up your database. If additional scripts are mentioned in *install-notes.txt* above, be sure to include the relevant scripts as well.
  - A set of full data files ending with *.txt*, one for each relation in your final project. The set must include all files needed for *setup.sql* to be run to completion on ugrad without errors. Note, if size limitations force you to hand in your data files via Piazza instead, please make a note of this in your Gradescope submission and ensure your files are all handed in before the deadline.
  - **setup-small.sql**, an up-to-date version of the *setup.sql* setup script above, but modified to create the tiny "testing" version of your project you submitted in Phase C. Any changes you made in your schema since Phase C should be reflected here. All additional setup commands used in the full *setup.sql* (for stored procedures, etc.) should also be present.
  - A set of shortened data files ending with *-small.txt*, one for each relation in your final project, so that *setup-small.sql* can be run to completion on ugrad without errors.

- a subfolder named **public_html**, which includes:

  - **JHED1_JHED2.html**, which is the main html file from which a user can interact with the copy of your database created from the scripts and text files submitted by you in the folder *dbase_setup* above.
  - All additional files required by the main html file named above to execute queries on your database and view the results. These may include, for example, additional .html files, .php files, etc.

- a narrative text file named **discussion.txt** describing the project, with required labeled subsections as follows:

- *Members* - the names and JHEDs of the group members who worked on this project
- *Modifications* - briefly summarize any changes you made to your project after selecting a domain and list of questions in Phase A (did you change subject domains due to lack of data? did you replace questions from your original list of 15, or expand the list? did you pull in an additional data source to enrich the project?)
- *Process* - an (updated, if necessary) version of process.txt from Phase C, explaining your data sources and how you cleaned and prepared the data for insertion into your database
- *Successes* - describe 2-3 technical portions of your project about which you'd like to brag a little (for example, what was a particularly challenging aspect of writing your database that you'd like to highlight so the course staff don't overlook its complexity?)
- *Known Issues* - a description of places in the implementation of your project, if any, where you did not fully succeed (are there bugs? are there unimplemented features?)
- *Extensions* - beyond fixing any known issues mentioned above, what additional functionality do you most wish you had time to add to your project? Describe at least two new features.

NOTE. If course staff can't properly install your database and/or execute queries from your html site, they can't grade your work properly. It is your responsibility to check that you have included all required files. Include EVERYTHING in this final submission, even if it is a file that hasn't been modified from an earlier phase. As you approach final submission, one partner should send the other the completed .zip file, and have the other partner test that installation executes without errors.

Contributions Report. Within 24 hours of your final code submission, each group member must submit a brief file named JHED_contributions.txt (use your own JHED please), containing:

1. Name: Individual's JHED ID and full name
2. Contributions: Short paragraph explaining how you and your group members worked together throughout the project (who took responsibility for what parts? did you meet regularly and code together, or just send each other your parts electronically?).
3. Above and beyond: If a member of your group deserves a "shout-out" for doing more than their share of the work, explain who and why. If no one does (i.e. your work was divided roughly equally, or), then simply write 'none'.
4. Other: Are there any other issues the course staff should know about?

Create this file separately from other group members' submissions. **Each member must turn their own file** (separate from the group's code submission), using a Gradescope link named *project_contributions*. Failure to turn in this file may earn the group member a 0 score on Phase E.

Phase E Final Interview. During the period of approximately May 11 through May 14, groups are required to attend a final interview with course staff (their assigned CA mentor and/or the instructor as well) to review and discuss the submitted work. The interview will take place over Zoom and will be recorded. Your assigned CA will be in touch to schedule this meeting. All group members must attend (with video on) and actively participate in the meeting. Schedule your meeting as early as practical given the schedules of all involved, but no earlier than the day after you submit your fully-completed work for Phase E. (In other words, you should only arrange your group's interview on May 11th if you're sure you will be handing in your completed work by May 10th.) Students who are expecting to graduate this semester should aim to schedule their meeting on or before May 12th if at all possible, and no later than the morning of May 13th.