

1. Reflecting on our final project as it is now, there were a few changes from what we originally envisioned in stage 1. For example, we intended to have many more types of statistics on different courses apart from average GPA or course ratings. However, this ended up being too tangential to the real purpose of the project, which was to create and manipulate a database, and ended up being scrapped in favor of a higher focus on making a better database and learning web development. Most other aspects of our project remained in line with our stage 1 report, other than forgoing the creative component because of a lack of time towards the end.
2. Thankfully, I believe our project did achieve the usefulness we hoped for. Namely, it merged relevant grade statistics with user reviews and ratings. Users can access the site and read a variety of reviews by similar students and clearly see average GPAs for each instructor. For more avid users, even more in-depth statistics could have been found utilizing much more complicated aggregation methods in mysql, but we could not accomplish this. What is there is still enough, however.
3. Our source of data remained the same from our first proposal, namely Wade Fagen-Ulmschneider's UIUC GPA dataset on his github (<https://github.com/wadefagen/datasets/blob/master/gpa/uiuc-gpa-dataset.csv>). Our schema was changed for some tables, though. Our Course_Offerings table had duplicates even with our different primary keys because of how the original dataset was formatted. Each course was divided by type of class too, like an in-person offering or online class. We aggregated the total grade data for these types of duplicates and added a "duplicate number" attribute to track it. Our Instructors table removed information regarding departments, since our dataset did not contain specific information of the instructors other than their names and thus it would not be practical to search through each one and add their departments. Further, when using specific courses, we could not find an easy way to determine their actual CRN, and instead used our own generated IDs as primary keys.
4. As a result of these changes, our ER diagrams would have been changed to include these additions or removals, but primary keys or foreign keys remained the same. Based on the information that could be gathered from the dataset, this was certainly a better design than our original schemas.
5. As was previously outlined, the functionalities we removed had to do with data aggregation and statistics. We originally planned on having much more of this, but settled on a simpler application due to the time we had to not just learn how to develop web applications, since none of us really knew how originally, but also how to integrate advanced SQL programs/queries into the application and display the values accordingly.

6. Speaking of the advanced programs, ours had to do with, for stored procedures, calculating the percentage of A's received in a class and calculating the average GPA per instructor per class. Our first advanced query helps with students understanding proportionately how many people attain the most satisfying score in a course, the treasured grade of "A". If this is high enough, a user would likely consider the respective course to be a good choice. However, for users who do not specifically wish for an A but rather a "high GPA", our second query would display the relevant statistic. The average GPA is also calculated by instructor, allowing for instructors to be considered specifically when deciding on a course. Each of these parts of the procedure create a new attribute to indicate whether they are good, not so good, ok, not so bad, etc. using ASCII faces just for fun or written words, and these are assigned depending on the values output by the queries in the stored procedure. We also had a trigger which, on adding a review, would calculate new course rating aggregates per course. This allowed for the courses to get their rating averages without a query but instead with the trigger since new reviews are constantly added by new users.
7. Regarding technical challenges in the project, this is what each of our team members had to say:

Safin Akash : The data must be thoroughly inspected before deciding the schema for the base tables. This is especially true for deciding the primary key because even if we logically decide on the candidate key, there might be special corner cases that allow for duplicate primary key entries in the data and the data would be rejected.

Nithin Balaji: It would be better to think of features for the website from a SQL perspective much beforehand. It was hard to implement the advanced query requirements at a later stage as we were thinking of implementing the features in a completely different direction.

Sandhya Rao: Rather than encountering major difficulties on the database side, we actually had a lot of setbacks when it came to implementing the front end interface and it was also too easy to be swept away by the technical aspects of the web, especially because all our team members did not know anything about web development prior to the project. So one thing that we could have realised earlier was to not let our focus shift to the web dev and instead focus more on just the database aspects of our project.

Sairam Penumarthy: One technical issue we faced was deciding on a proper database structure and schemas. The nature of our project at the start made it so not many tables were needed, and thus led to a lot of time spent trying to expand our project at the last minute because of the 5 relational table minimum requirement. In hindsight, I would have first read ahead to future

project stages to make sure our original idea could easily fit into later required specifications instead of developing additions much later in the course.

8. There were no further changes other than the ones outlined above.
9. If we were to revisit this project to further improve it, we would probably find a way to add more info into tables such as Instructors, which did not have information on instructor departments, and Course related tables by finding the CRNs of previous courses so students would have an easier time searching for the course they want using CRNs.
10. Finally, our work was divided quite evenly. Nithin implemented the user authentication and logging in part of the project and a trigger for aggregating review stats; Sandhya implemented the find/post review section of the project, essentially ensuring all the calculated stats and reviews are plugged in for a holistic overall flow in the website; Safin did a lot of cleaning, importing and designing the database and a procedure for finding difficulty level of review based on gpa percentage; and Sairam made the basic foundation of the project website along with data-plugin mechanisms and a stored procedure. Our teamwork was managed well by meeting at the Grainger library to coordinate on our project in-person.