

**CSD310: Database Development and Use**  
**Module 12.3 Assignment: Lessons Learned**  
**Isaac Ellingson**  
**12/20/2025**

Overall I would call this project a success. Not everything went right, not everyone is going to be happy, but I know that all of us that stuck with it and turned out a presentation are proud of what we put together.

First point I need to make, I'm not a natural team leader, but I am. My schedule's more flexible than most, I have strong opinions / project vision, and after getting to know me, people are generally comfortable taking my lead. In group settings, I'm very proactive about work. I want to know what's happening when so I can plan my week. For these reasons I wind up in leadership positions a lot, as I did here. But... I find the moderation and conflict resolution roles to be challenging. Thankfully this doesn't come up a lot, but it's a little ironic that people are my achilles heel here. If I can keep a team aligned and invested in the product, we're golden.

Second, in in-person group projects, I have never in my life seen someone ghost the team. Somehow I see this a lot in online classes. I'm likely one cause, I know I'm a challenge to work with. But online groups are also less of a family system like an in-person group, they're much easier to disappear from, and I can imagine a variety of reasons to disappear. For the record, if someone ever says "I can't work with Isaac", let's get them switched with no fuss, and then I know and can plan around it. Whatever the case, we lost Jeffrey somehow, but the show must go on.

We were completely unanimous on the Winery project, and the ERD practically wrote itself. Patrice gave a lot of good naming notes. I think things may have been challenging for Jose because it looks like at times he had to connect from a phone, but he pulled his weight. I definitely have no complaints about the team.

Some friction occurred at Milestone #2, where Jeffrey did the initial SQL implementation, which had numerous differences from the ERD. Jeff wasn't available during the earlier parts of the week when we were making the design, so I would have liked to have had a discussion about tweaking some aspects of the design to match the SQL, instead of the other way around. He refused to engage about the design, so I dropped it. On the plus side, the work he produced included numerous CHECK constraints – which MySQL used to completely ignore. I was pleasantly surprised to learn that it enforces them now, so we have a database that aggressively rejects invalid data. (This caused problems later, because we couldn't really fudge the hours data, and yet I regret nothing)

As the milestones moved on, we found scheduling that worked really well for us, and rotated out which person assembled and submitted work. The team really was a well-oiled machine. In the past I definitely would have, say, tried to put together the whole initial SQL, or the whole slide deck, or both and left nothing to chance. As I get older I'm learning that less is more, and it's always great to unexpectedly see slides with pictures of a real winery, or check constraints that actually work, or unexpected columns with counts of late vs on-time deliveries that really make a report pop.

Tools also worked out great. Discord meant that we could do hybrid online/offline work, post work as soon as we're satisfied with it, pop up with impromptu chats before a scheduled meeting, and share and re-share artifacts in a review process. Involving GitHub also made sharing easy – we didn't use it a lot, but there were opportunities to demonstrate changes, make suggestions, and combine work with that. At my suggestion, we assembled our presentation notes using HackMD, and I'm so happy that everyone got involved in that. It produced a really strong presentation, and the live nature of the session gives me confidence that no LLM junk was able to slip in under the radar. This in turn produced a markdown artifact that you'll see saved in our repositories.

The main thing I “took over” was the python code this time. I had already written code to take a cursor, figure out column labels and widths, and print out a result set regardless of what the data happened to be. Reusing this, custom code for each report was completely unnecessary. I also took care of assembling the final SQL script with the data each milestone – this way I could ensure an access-restricted database user was present, and I felt that I was best equipped to deal with how big this thing was going to get. I think I was right. It gave me challenges worth my time, and tapped the brakes a little on the difficulty level for everyone else. I'd do it again this way, for sure.

One thing that went wrong, at least momentarily, was that we met to discuss feedback, created a list of changes needed, did the work and put together a final presentation, and *then* saw your post in our group forum. This is not ideal. We decided to delay the project another day to make sure we had adequately addressed these points.

This is good exercise for me. I do code review and mentoring all the time, but this is a very different setting, where everyone is expected to do a similar amount of work on a project they did not necessarily choose for themselves. I think it's incredibly likely that I'll run a little software shop some day, and this is much closer to the circumstances I'm going to encounter there. My style may be unconventional, I don't seem to click with everyone, but I'm really happy with how focused and organized this team was. I'm open to feedback, but I'm going to count this one as a victory.