CS 325 - Project Handout - Fall 2024

When studying the design and implementation of databases, it greatly helps understanding to actually experience the process being studied. So, during this semester you will be modeling, designing, and implementing a database for a scenario. (Think of it as modeling, designing, and implementing a **demonstrable prototype** of a database for that scenario.)

This is a group project. You are required to form a group of **three** members. Working on this project as a team is essential because database design is a collaborative process that mirrors real-world scenarios. Each member of the group can bring different perspectives and ideas to the table, helping to ensure a well-rounded and comprehensive project outcome. By working together, students will also develop essential teamwork skills, learn how to divide tasks, and experience the collaborative nature of building a robust database model. Group-based projects promote problem-solving, effective communication, and an exchange of ideas, which are critical when designing and implementing databases that meet real-world requirements.

This project will use the Oracle DBMS (Database Management System) on campus. The project consists of **several milestones**, with pieces turned in for each.

Project Milestones

- the overall project grade -- the sum of the parts below -- makes up **15%** of the final course grade, as noted in the syllabus. (this sum is multiplied by 0.15 to compute the project portion of your semester course grade.)

Project Milestone	Milestone's Worth	Additional Information
Scenario Selection	up to 5 points	due **11:59 pm on Monday, September 30**
Project Model Draft	up to 17.5 points	due **11:59 pm on Wednesday, October 23**
Project Design Draft	up to 15 points	TBA (To be announced)
Project Population	up to 7.5 points	TBA (To be announced)
Project Presentation	up to 5 points	TBA (To be announced)
Project Final Milestone	up to 50 points	TBA (To be announced)

Project Grading Comments

Your final project should be robust enough to **demonstrate** as a **prototype** -- do not attempt to implement a "production quality" system! (And if you cannot implement everything you hoped to, implement as much as you can, aiming for an **interesting, demonstrable prototype**.) A project that meets all of the minimum requirements mentioned in this handout, and that has met all of the minimum requirements all down the line (throughout the milestones), and that does so well, would receive a grade of **90**.

The other **10** points I will reward based on merit -- did you come up with a particularly interesting, original proposal? Is a database particularly well-designed and implemented? Is something about a project above-and-beyond the minimal requirements, or exemplary? Were useful extra features included, or was some major aspect particularly well-done? Did something about a project just stand out, or make a strong impression?

Especially for the project model, but also for the project design and population, you **should** likely be **improving** them over the course of the project. So, you usually are required to submit the **latest** versions of files from **earlier** milestones as part of **later** milestones. This is a requirement, and your grade will be affected if you do not include these.

Project minimum structural requirements

Because an important topic in this course is database modeling and design, your project is required to meet certain criteria to help ensure that it is at least somewhat interesting in model and structural senses. **Final projects that do not meet these minimum criteria will be severely penalized.**

The **database model** for your database must include:

- at least 10 **distinct**, **significant** entity classes
  - a (non-union)-superclass entity along with all of its subclass entities count as **one** combined entity toward the ten-entity-class minimum, unless some of the subtypes have relationships in which only that subtype participates. Be careful, and **ask me** if you have concerns about this.
- at least 8 **distinct**, **significant** relationship classes, at least two that is **1:N** and at least two that is **M:N**
- at least two **multi-valued** attribute

And, the corresponding **database design/schema** must eventually **correctly implement** all of the above.