

Final Project

Database Design and Development (CS4092) – Summer 2025

Project Overview:

This project gives you hands-on experience designing and building a relational database system. You will go through the complete database development lifecycle, applying the concepts learned in this course to develop a functional backend database using a relational DBMS such as MySQL or PostgreSQL.

Project Phases & Deliverables:

1. (10 points) Requirements Gathering:

- Identify and document the data requirements and use cases for your system.
- You may define your own system, use publicly available enterprise requirements, or adopt the default e-commerce project below.
- Deliverable: Requirements document (PDF).

2. (15 points) Entity-Relationship Diagram:

- Create an ER Diagram representing entities, relationships, attributes, and keys.
- Tools like Lucidchart, draw.io, or dbdiagram.io may be used, or you may draw by hand—just ensure the drawing is readable.
- Deliverable: ER diagram (image or PDF).

3. (15 points) Schema Design:

- Convert the ER diagram into a relational schema.
- Ensure clear primary/foreign keys and relationship tables where needed.
- Deliverable: List of relational schemas (can be in a PDF document).

4. (15 points) Database Implementation:

- Implement the schema in any relational DBMS of your choice.
- Populate the tables with sample data using SQL INSERT statements.

- Deliverable: SQL script (.sql file) containing schema creation and insert statements.

5. (45 points) Database Interaction:

- **(15 points) SQL Queries:** Write at least one multi-table query. Example: Show the names of customers along with the names of the products they purchased where product price > \$100.
- **(30 points) Business Logic:** Write code in a language of your choice (e.g., Python, Java) to demonstrate application-level interaction with the database. This can be through a terminal/console interface.
- Deliverables:
 - Video demonstration of SQL and code-based interaction.
 - Source code included in GitHub repository.

Group Scaling Expectations:

- Individual projects must include at least 4 relations (tables). Simple CLI-based business logic is sufficient (e.g., inserting new products into the database, viewing products).
- For group projects, each additional member increases the minimum relation count by 1:
 - 2 members: 5 tables, with CLI-based business logic including more comprehensive features such as product viewing based on search criteria.
 - 3 members: 6 tables, same as above but with additional CLI features.
 - 4 members: 7 tables, with CLI-based business logic similar as above and a basic front-end implementing selected functionality (e.g., inserting into a table). Not all features require a front-end.
 - 5 members: 8 tables, with more comprehensive functionality and broader front-end support for selected features.

Sample Project (Optional):

You may optionally choose to build a backend for an e-commerce system, or adapt it to another domain (e.g., library, booking, inventory).

The e-commerce system should support two main user roles:

- Staff members: Responsible for adding/editing products, managing inventory, and possibly overseeing transactions.
- Customers: Able to browse available products, add multiple credit cards to their account, and make purchases.

A basic schema for this project might include the following five core relations:

- Customer
- Product
- Staff
- Purchase
- CreditCard

To increase complexity, especially in group projects, consider incorporating additional entities and relationships, such as:

- Return (to handle returned products)
- Shipment or Delivery
- Seller or Distributor roles
- Review or Rating for customer feedback

You are encouraged to scale your design based on group size, increasing the number of entities, the complexity of relationships (e.g., many-to-many or ternary relationships), and the variety of queries and logic implemented.

Submission Requirements:

- A video walkthrough that demonstrates your project and explains key features.
- A GitHub repository with your source code and documentation.
- Only one group member should submit to Canvas. Be sure to list all group members in either the video or submission description so everyone receives credit.

Extra Credit:

- Up to 10 bonus points for active use of GitHub throughout the development process (frequent commits, branches, issue tracking, etc.), not just a final upload.

Important Notes:

- If you're unsure whether your project idea fits the expectations, feel free to run it by me—even informally. I'm happy to help brainstorm, clarify scope, or make sure you're on the right track (especially for group projects).
- You're welcome to integrate this project with your Programming Languages project if your application logic uses a language from that course. If you choose this route, your project should include a meaningful focus on either the business logic or front-end interaction to ensure it's substantial for both courses.