



Database Project Instructions

(This document will update frequently, so keep watching for updates and changes.)

Overview

Throughout this course, you are learning the skills of a database designer and developer that leverages the features of relational databases. Typically you start with a domain and design the entities, attributes, and relationships. Then we progress from the entity relationships to the relational schema. Finally, we implement the design using SQL and have a running database that represents the design that is driven by the domain.

For your course project, you must demonstrate your command of these course topics by creating a modest database around some domain of your choosing. It's easy to choose a domain that is large and complex. Please refrain from too many tables and attributes.

This project can be viewed as three related projects with an E-R Design, a Relational Schema, and a SQL Implementation.

E-R Design

Choose a business or nonprofit organization to model. At the end of these instructions are several ideas to spark your creativity. Do not choose the domain we discussed in the classes.

Your design should include at least 8-10 entity sets, including at least one weak entity set. There should be at least one composite and one multi-valued attribute. There should be at least one many-many relationship and one descriptive attribute; you will probably have a number of many-one or one-many relationships. Be sure to indicate any requirements for participation, as well as the cardinality of relationships. Your design can be presented as a hand-drawn diagram, as long as it is clearly legible.

Relational Schema

Convert your design to a set of relational schema. Be sure to indicate primary keys. Verify that your design is in BCNF, or if it is not, decompose it so that it is. (There is no preference for which of these is the case, but you should show your work either checking that it is in BCNF or the process of decomposition to make it so.)



SQL Implementation

Your database design shall be implemented with SQL files. In addition to primary and foreign key constraints, including at least one check test. In order to write queries and test them, you'll need to populate your tables with data. There were several examples of SQL files for the University, Employees, and Banking databases that you can reference to create your own database. All SQL statements must be in SQL files that can run in SQLite. Document the intent of each command or group of commands directly in the files as SQL comment. Provide one SQL file for the DDL, one file for the DML, and another file for the queries.

Query Requirements

Each query should be accompanied by an English statement of the intended result of the query. For instance, in some commercial world domain that involves shipping products, you might write "Give the total, in weight, of each product shipped during the week of July 4th, 2023."

- ☐ At least two queries should involve four or more relations
- ☐ At least one query should involve outer joins
- ☐ At least one query should use an aggregate function
- ☐ At least three queries should use subqueries in a non-trivial way
 - ☐ One of those should use a set comparison (e.g. > **some**)
- ☐ At least two queries should use grouping
 - ☐ At least one of those should use **having**
- ☐ At least one query should use set operations

Your project work will work through these stages

1. Choose a domain and scope
2. Create the E-R Model
3. Create the Relational Schema
4. Document normalization with BCNF
5. Create the database using SQL
6. Seed database using SQL
7. Provide the requested SQL queries.



Submission

Due May 1.

Submit your work to a GitHub repository. Make the repo private and add just me as a collaborator. Add your repo to Moodle as a submission. My GitHub ID is “javajon” and email is jonathan.johnson@dijure.com. The repo should contain the report and SQL files.

- **Report:** Use the provided project report template and submit the following in a single PDF or readme.md file. A readme.md file is ideal for GitHub. The report shall include
 - ☐ Your name
 - ☐ Your domain description summary (a paragraph or so)
 - ☐ A description of the scope limit, assumptions, and other ideas you may have considered but was not included in the scope
 - ☐ E-R model diagrams. If drawn, scan in and avoid wonky camera images.
 - ☐ All Relational Schemas as visual models or parenthetical expressions
 - ☐ Normalization steps with BCNF
- **SQL:** All working SQL files that must run in SQLite
 - ☐ A SQL DDL to set up the database from scratch
 - ☐ A SQL DML to seed the database with data
 - ☐ At least 6 queries



Plagiarism and academic dishonesty

This is not a group project. Work individually.

You are encouraged to consult with one another when you work on homework assignments and programming projects, but in the end, everyone must do their own work to hand in. In particular, discussion of assignments/projects should be limited to brainstorming and verbally going through strategies, but it must not involve one student sharing written solutions with another student. Everyone must write up solutions independently. If you have discussed with your classmates or used any outside source, you must clearly indicate so on your solutions and provide all references.

⚠ Turning in another's work under your name is plagiarism and qualifies as academic dishonesty. Academic dishonesty is a serious intellectual violation, and the consequences can be severe. For more details, read the [Student Handbook 2022-2023, pp. 4, 29](#)

Grading

E-R portion:

Required elements	20%
Logic	5%
Thoroughness	5%
Clarity	5%

Relational schema

Correctness	25%
BCNF documentation	5%

SQL DDL implementation	20%
Queries	<u>15%</u>
	100%



Some domain ideas for inspiration and guidance

Please avoid domains we have already discussed such as the University, Banking, and Employee domains. Let your imagination run when considering your choices of domains.

E-commerce	Virtual interior design
Online marketplace	Video game collection tracker
Social media	Medical appointment scheduling system
Fitness tracking	Online marketplace for handmade goods
Recipe sharing	Charity fundraising platform
Job board	Movie recommendation engine
Customer relationship management (CRM)	Online therapy platform
Project management	Social network for travelers
Inventory management	Music festival management
Online education	Humanitarian aid distribution
Travel booking	Fashion e-commerce site
Healthcare management	Virtual art museum
Music streaming	Mental health support community
Video sharing	Sustainable energy management
Online gaming	Recipe sharing for special diets
Real estate	Social network for language exchange
Event planning	Online marketplace for vintage goods
Online survey tool	Environmental protection database
Customer feedback	Music instrument inventory management
Online appointment booking	Online resource library for teachers
Restaurant management	Personalized nutrition planner
Online auction	Charity auction platform
Car rental	Public transportation management
Home automation	Cryptocurrency portfolio tracker
Online booking system for a hair salon	Sports betting platform
Online delivery service	Online craft marketplace
Sports team management	Wine cellar management
Stock trading	Gardening planner
Time tracking	Virtual wardrobe organizer
Freelance marketplace	Online book club
Pet adoption	Personal finance management
Virtual reality gaming	Social network for hobbyists
Fantasy sports league	Non-profit donor management
Online art gallery	Smart home automation
Dating app	Social network for pet owners
Language learning	
Recipe generator	