Principles of Data Management CSCI 320
Recipe Domain

v3.1

1 Project Description

This is a semester long project.

You will be required to create a database based application in a team. You will be given a choice of application domains. This document details the requirements for the recipe domain.

This domain will require that you create a database to manage recipes and ingredients.

The project will be broken into 4 phases. Each phase will build on the prior phase. As basic description of each phase is outlined here. More details in the remainder of the document.

- Phase 1: Select a domain and identify the type of application you are going to write. Due: June 6, 11:59 pm.
- Phase 2: Generating an EER model for the approved domain. Due: June 22, 11:59pm.
- Phase 3: Generating data and loading it into the database and normalizing it. In addition, you must provide an interface to your dataset to manipulate the data. Due: July 13, 11:59pm.
- Phase 4: Performing analytics on your database, generating a report and poster with the results and an updated video. Due: August 3, 11:59pm

Feedback will be provided for each phase. You can select to receive a grade on a phase before its due date so you can move on to the next phase. **Once you select this option the grade becomes final**.

2 Data Requirements

- Multiple users can be managed by this system. The `user` table must has at least the following attributes:
  - Username (`not null, unique`)
  - Password (`not null`)
  - Creation date (`not null`)
  - Last access date (`not null`)
- Users will be able to create recipes. The `recipe` table must at least has the following attributes:
  - Name: The name of the recipe (`not null`)
  - Description: A short description of the recipe (e.g. A festive fall fruit salad made with yogurt instead of the usual mayonnaise)
  - Servings: The number of serving the recipe makes
  - Cook time: The amount of cooking time this recipe requires (in minutes)
  - Difficulty: The difficulty of the recipe. (`not null`)
    Domain: Easy, Easy-Medium, Medium, Medium-Hard, Hard. (`default value Medium`)

  - Rating: A rating from 0 to 5 for this recipe
  - Categories: The categories this recipe belongs to. A recipe may be uncategorized
  - Ingredients: The ingredients required for this recipe (`not null`)
  - Steps: The steps to complete the recipe (`not null`)
  - Created by: User who creates the recipe (`not null`)
  - Creation date (`not null`)
- Users can keep track of the ingredients they have (aka pantry). Your database must be able to store at least the following information about each item in the `pantry`:
  - Item name (`not null`)
  - Quantity bought (`not null`)
  - Current quantity
  - Aisle (e.g. baking goods, canned and jar food, dairy, meat, etc.)
  - Purchase date (`not null`)
  - Expiration date

3 Group selection

Self enroll in a group in myCourses. Groups may be no more than 4 people, but may be cross section. Students not enrolled in a group by the due date will automatically be assigned a group.

4 Phases

Below outlines the minimum submission requirements for each part of the project. Submissions are due by 11:59 pm on the day they are due. No late submissions will be accepted. No email submissions will be accepted.

4.1 Phase 1

During this phase your will begin to outline your project design. Submit a 1-page pdf report about your project including the following information:

- Default group number from myCourses
- Team Name: You must choose a new name for your team
- The names of all team members
- Your selected domain, and a backup domain
- The description of your project including your approach, how you intend to provide an interface to your dataset (command line application, web app, desktop app), and the front end language you plan to use. This description must be presented in an Introduction section; you will continue to fill in this report in future phases.

Submit a PDF of the report outlined above, named **Phase1.pdf**. Submission of anything other than a PDF will earn a 0 for this phase.

MyCourses will only accept one file and only save the last submission.

Your selected domain will be granted on a first come, first serve basis. Only 1/3 of the teams in any given section will be granted a domain. For that reasons, your domain must be approved before proceeding with the remainder of the project. You can submit early (and email the instructor) to verify approval before the due date, though once it is approved, it becomes final. Your approved domain will be listed on myCourses under groups. For example, if your domain is Widgets, your team name in myCourses would be "SQL Injection (Widgets)".

4.2 Phase 2

During this phase you will begin to design your database system. This phase is a complete design for the rest of the semester. The goal is to create a conceptual data model using the EER model for your given domain from Phase 1. The following items are due:

- An update to the phase 1 report:
  - details the decisions that were made for the EER diagram
  - explains how the reduction to tables were done for each of the entries in the EER diagram
  - information in the report must match the EER diagram and reduction to tables submitted
- An non-hand drawn EER Diagram that:
  - uses the proper EER notation (ie. rectangles for entity types, underlined primary keys, etc)
  - depicts the entity types from your domain (for all phases)
  - properly uses weak entities, specialization, etc. (ie. there must be a reason for the specialization; special relationship or attributes)
  - depicts the relationship types between the entity types. The relationship types must enforce the data requirements and program operations of your domain
  - has proper cardinalites. Cardinalities must make sense with the requirements provided and uses the correct notation
  - properly uses attributes types on entity and relationship types; including key attribute types, derived attribute types, etc.
- Reduction to tables that:
  - uses correct notation for reduction to tables
  - matches EER diagram (i.e. each entity type is an independent table, a multi-value attribute in the EER diagram is stored in its own table)
  - handles cardinalities properly (i.e. many-to-many relationships reduce to an additional table)
  - every table has a primary key
  - foreign keys are properly added where needed

Submit a Zip file containing 3 files, the report outlined above (PDF), the ER Diagram (PDF), and a Reduction to tables (PDF), named **Phase2.zip** Submission of anything other than a Zip containing PDFs will earn a 0 for this phase.

EER diagram must be easily readable in the PDF.

MyCourses will only accept one file and only save the last submission.

4.3 Phase 3

During this phase, you will load your data into a real database, based on the EER diagram and reduction to tables previously submitted (corrected based on any feedback). You will also submit your database application program, as well as a 3-7 minute video of the application running.

4.3.1 Application Requirements

- Users will be able to create new accounts and access via login. The system must record the date and time an account is created. It must also stored the date and time users access into the application.
- Users will be able to create, edit, and delete recipes. Users can only modify or delete recipes created by them. A recipe cannot be deleted if another user has already made it.
- Users can organize recipes into categories to make them easier to manage. Users can create their own categories (e.g. desserts, breakfast, seafood). A recipe can belong to more than one category
- Users will be able to search for recipes by ingredients, name or category. By default, the resulting list of recipes will be sorted alphabetically (ascending) order by name.
- Users can sort recipes by name, rating, or most recent (ascending and descending).
- Users can mark a recipe as made only if they have all the recipe's ingredients and quantity. The system must record the date the recipe was made.
- Users are able to scale the ingredients of a recipe when making the recipe. The amount to scale can be entered as a whole number, or decimal (ex: 3, 0.25).
- Making a recipe will reduce the quantity of the item in the pantry. Users can also update items quantities manually (outside the scope of making a recipe)

4.3.2 Submission Requirements

- An update to the phase 2 report:
  - outlining any changes to your EER diagram and reduction to tables since Phase 2
  - contains sample (3-8) SQL statements used to create your tables
  - contains sample queries (2-5 for each table) for populating the data
  - contains a description of how the data was loaded into the database (was the source manipulated, did you use a tool, etc) with 5-10 sample insert statements
- An EER Diagram (even if not updated)
- Reduction to tables (even if not updated)
- Large amount of data loaded into the postgres server that matches the reduction to tables and the EER diagram, see below for what is considered a large amount of data.

- Data in the database should be normalized to 3NF
- Part 1 of your application program. At this point, your application must be able to perform all the basic CRUD operations stated in the **Program Operations - Phase 3** section of your selected domain. Submit your source code (in a src folder)
- A 6-10 minute video demonstrating the application running and manipulating data with a voice over explaining the application. We will not be running your application, it is important that your demo covers the functionality of the application.

Data must be loaded into the proper account area, not the public postgres user area. Failure to load data into the proper location will result in a 0 for this phase. If there are questions, please ask.

Data in the dataset may have been converted to numbers for storage. (ie. 1 for male, 2 for female). You must store them as their non-numeric values (ie. male and female).

You must have enough data in this phase to do phase 4 and have it loaded for this phase. How much is enough? 10s-100s of rows in each table, while a M:N table should have 200-500. To simplify, after a join with a complex query, you should have several thousand rows.

Submit a Zip file containing the report outlined above (PDF), the ER Diagram (PDF), and a Reduction to tables (PDF), and your source files named **Phase3.zip** Submission of anything other than a Zip will earn a 0 for this phase. All 3 PDF files should be in the root of the zip.

MyCourses will only accept one file and only save the last submission.

4.4 Phase 4

For this phase, you must complete your application program including the recommendation system and some extra functionalities. You must also perform some data analysis to discover useful information.

4.4.1 Application Requirements

- Users can rate a recipe made. The rating value must an integer value between 0 and 5 (both inclusive)
- The application must provide a recommendation system with the following options:
  - Top rate recipes: The top 50 most recommended recipes order by rating (from high to low)
  - Most recent: The 50 most recent recipes order by creation date (from most recent to less recent)
  - In the pantry: It will list recipes based on the items in the pantry. A recipe will be listed only if the user has all the ingredients and the right quantity to make the recipe. The recipes will be sorted by rating (from high to low)
  - Recommended to you: Recipes made by other users who make the same recipes. The recipes will be sorted by rating (from high to low)

4.4.2 Submission Requirements

- Final version of your application program. At this point, you application must include all the functionalities stated in the **Program Operations - Phase 4** section of your selected domain. Submit your source code (in a src folder)

- An update to the phase 3 report:
  - explaining the process/techniques used to analyze the data (what types of algorithms were used, did you use a tool for analytics, or did you create materialized views, etc)
  - explaining the indexes created to boost your application program's performance
  - containing an appendix listing all of the SQL statements used in this phase

- A poster showing:
  - your team name
  - the names of all team members
  - the observations from the data analytics
  - technologies used (Excel, Python, etc)
  - visual representation of the data (charts, graphs, and other visual representations are required)

  Designing an effective poster:
  - include the team name and the name of all team members at the top
  - keep any text brief
  - do not use all capital letters
  - user graphics (charts, graphs, etc) that can be understood in one minute or less
  - add a descriptive caption below each figure; table heads should appear above tables. Use the abbreviation (e.g. "Fig. 1") at the beginning of each caption. That will make easier to refer to those figures during your poster presentation.
  - have 3-4 graphs/charts explaining the data with brief descriptions. These should not be trivial.
  - use color, sparingly. Find a color palette that works and is not distracting.

  Poster must be easy to read and understand. The viewer should gain new knowledge or insight by just looking over your poster. You can find a poster sample at myCourses. You can also can take a look at the CS MS Capstone project's posters. To access to that information, you will need to login with your CS account.

- A **single** 7-10 minutes video (or link to a video) that demonstrate the final version of your application and present your poster. Your video must be structured as follows:
  - Start by demonstrating the final version of your program. Make sure to demo all functionalities defined in the **Application Requirements** section of this domain. During the demonstration, you must show from your source code, at least two complex queries (e.g. multiple joins, nested queries, correlated queries) implemented during this phase. We will not be running your application. It is important that you demonstrate all required functionality.

    - After that, you will present your poster.
        - Familiarize viewers with the fundamentals of your program quickly and easily
        - Use the information from your poster to present only the highlights
        - Explain the patterns or conclusions drawn from the data analysis

The type of data analysis you will perform is up to you. Here you have some ideas:

- use analytics tools (e.g. Weka, R)
- perform Exploratory Data Analysis (EDA) such as
    - export your data into Excel to determine interesting information and generate charts (e.g. bar chart, pie chart, histogram, scatter plot, correlation matrix)
    - time series analysis
    - descriptive statistics
      Notice that you are allowed to add some descriptive statistics in your report and/or poster, but your data analysis cannot be solely conform by this type of EDA

Note that you can also perform descriptive and/or predictive analysis of your data but it's not required for this course. **Important!** Your analytics must not be trivial (e.g. finding that people like pizza).

Submit a Zip file containing the report outlined above (PDF), a Poster (PDF), the EER Diagram (PDF), and a Reduction to tables (PDF), a Video (mov or other movie file) and your source files in a file named **Phase4.zip**. Submission of anything other than a Zip will earn a 0 for this phase. All PDF files should be in the root of the zip.

MyCourses will only accept one file and only save the last submission.

5 Peer Evaluation

You must submit a peer evaluation for yourself and your team members. Failure to submit a peer evaluation will result in a 10% deduction on your project grade. A template is available on myCourses.

6 Datasets

There are plenty of free datasets available that you can use for your project. They may not contain all the information that your project required but you can combine several datasets together and/or you can also generate synthetic data. Find below some datasets you might find helpful:

- https://www.kaggle.com/shuyangli94/food-com-recipes-and-user-interactions

You can find more datasets in Kaggle or using Google dataset search engine.

7 Project Constraints

This section outlines details about any project constraints or limitations.

Constraints/Limitations:

- You must use a provided domain
- The constraints `required` and `unique` defined in your selected domain must be enforced for the database system by the definition of integrity rules.
- There may be also some attributes with an specific domain. That domain must be also enforced by creating integrity rules in the database system (e.g. the difficulty level of a recipe, the status of a borrow request, the rating value)
- The use of an ORM is prohibited. You must write your own SQL statements for phases 3 and 4. You are welcome to use a tool for importing data and creating the tables.
- Your data must be loaded onto the CS Postgres Server (reddwarf)
- Data analytics must be done on the data loaded, not the original source
- Your analytics must not be trivial (e.g. finding that people like pizza)

8 Grading

Your implementation will be grading according to the following:

- 10%: Phase 1

- 25%: Phase 2
  Total points: 100
  - 15%: Report
  - 55%: EER Diagram
  - 30%: Reduction to tables

- 30%: Phase 3
  Total points: 100
  - 20%: Report
  - 10%: EER Diagram and reduction to tables
  - 30%: Database implementation and data population
  - 25%: Application Program
  - 15%: Demo video

- 35%: Phase 4
  Total points: 100
  - 30%: Report
  - 30%: Application Program
  - 20%: Demo video
  - 20%: Poster

General deductions:

- -10% - Incorrect zip (must be flat except source folder)
- -20% - Report not an update from previous phases
- -100% - Submission of anything other than a Zip file

9 Submission

Follow the submission instructions for each section of the project. Emailed and late submissions will not be accepted.

FAQ

This section contains common problems and solutions when working on this project

Why am I unable to connect to the database at peek times?

Make sure you close the DB connections. While they will eventually time out, it is important you close your connections the same way you'd close files.

How should we compute analytics? You did not provide details on the queries to write.

You are welcome to be creative in this phase. The important part is that the queries are not trivial (such as users increase over time). There should be something that you wouldn't say, "of course they did".

Do we need to deal with security?

No. Passwords can be stored in plaintext. In a real project, these should be encrypted, but is not required for this project.

Can we see an A-quality poster?

No. We used to provide an A-quality poster and every poster turned in looked exactly like it. Take a look at the Master's projects. You'll know which are good.