Check Your Plate: A Nutrition Database Application for Dietary Management

A Report for Final Project of Course
[332:569] Database Engineering
Instructor: Dr. Dov Kruger
Rutgers University

Git Repo: https://github.com/onlinezyc/ECE-569-Final-Project-Check-Your-Plate-Yujia-Jiayi

by

Yujia Cheng (yc607) yc607@scarletmail.rutgers.edu
Jiayi Zhang (jz998) jz998@scarletmail.rutgers.edu

May 6th, 2024

Introduction:

Check your plate is a database application with an user interactive web front-end. Check Your Plate offers a personalized and comprehensive dietary management platform, utilizing a web-based solution to generate recipes and provide nutrient summaries. It serves to provide personalized diet solutions by leveraging a structured backend database and dynamic frontend interface. This approach enables efficient data processing to deliver relevant and safe dietary recommendations. We believe our project stands out by offering tailored solutions that address the complexities of dietary planning and allergen management. Its interactive web interface allows users to engage with nutritional data dynamically, offering insights superior to existing solutions. Furthermore, various existing dietary management solutions currently exist but lack the personalized and dynamic approach Check Your Plate offers. Prior applications struggle with accurately incorporating user preferences and providing effective allergen alerts. Check Your Plate bridges these gaps with a robust data model and flexible user interface.

Technical Details:

We first constructed a MySQL database that contains information of 5,911 foods and their respective individual nutritional composition. The source of the food-nutrient data is from USDA foodation food and survey food report. Our constructed database enables efficient CRUD operations on food items and nutrients. Furthermore, users of our database can create recipes out of existing food items, and the nutritional content of an entire recipe can also be easily queried using stored procedure.

1. Database Design

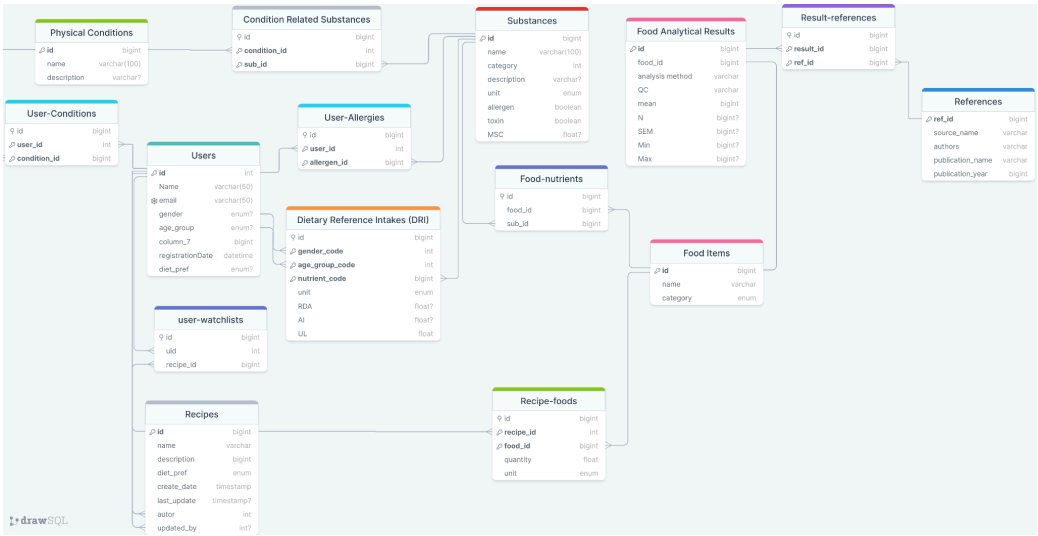

Fig 1: Check Your Plate Database Entity Relation Diagram

The database schema is designed per the services the final project should provide, with scalability and extended features in mind. The ER diagram for the Check Your Plate database can be seen in **Fig 1**. There are three core tables, which are 1) the Foods table, 2) the Substances table and 3) the Users table. Since the primary goal of our project is to deliver efficient food nutrient content query results, proper storage for food data and nutrients data is a bare-bone minimum. The Foods table stores information related to food, and it contains three columns: id, name of the food, and any description of the food. The substances table stores data related to any substances that are found in any food. Aside from the standard id, name and description columns, the substances table also contains unit, allergen, toxin and MSC columns: the unit column is for storing a standard unit used to measure a specific substance; the toxin and allergen column both stores boolean types, they serve as indicator signifying weather a substance is a known allergen and or a reported toxin, and the MSC, short for Maximum Safe Concentration, stores a float value meaning the maximum dosage of a substance that's reported to be safe for human consumption. By design, the MSC column needs to be filled only for a substance with toxin set as true. Since many foods can contain many substances and vice versa, the direct relationship between a Food entity and a Substance entity can be described as many-to-many. Normalizing many-to-many relationships is crucial in database design because it ensures data integrity, consistency, and efficient data management. Thus, a combined table Food-nutrients is adapted to flatten the many-to-many relationships into two one-to-many relationships.

The Users table is designed to contain all the users' information, and all the other user related features need to refer to the Users table. The most primary feature of our project that's related to users is the recipe building utility. Thus the Users table is connected to the Recipes table with the user-watchlists table, which serves as a combined table to normalize many-to-many relationship between user entities and recipe entities. Similarly a recipe-foods table is used to normalize the m-m relationship between recipes and foods.

2. Data Cleaning

The bulk of the project relies on data acquisition. The source USDA food data was quite large: totalling 98 MB of .json entries. Significant amount of effort went into understanding the source data structure, its content, and how to best extract the necessary information from the source and transfer the digested data into the MySQL database.

By studying the source data and referencing the designed database schema, the source data was processed to produce more condensed data (food_info_digest.json) which can be better adapted to the MySQL tables. This data trimming process effectively reduces the size of incoming data by 25%, from 98MB to 73MB.

Two approaches were proposed for transfering the data from the digested json file into the Check Your Plate MySQL database: 1) combining data cleaning and data parsing as one process, and 2) separating the data cleaning process and the data parsing stage. Over encapsulation does not

improve data transfer efficiency in this case, as the first approach suffers from constant transfer disconnection and rollback due to various errors. The second approach is more reliable and efficient as the edge cases that can cause errors were caught in the data cleaning process. Overall the data transfer process using the second approach was optimized to under five minutes, and resulted in a total of 5199 food records, and 222 substance records registering into the Check Your Plate MySQL database.

To summarize, the source USDA data was first trimmed and parsed into a condensed .json file, then the condensed data was transferred into MySQL database using mysql connector. Both the processes were carried out using Python.

The Python scripts used for the data trimming and data parsing can be found in the git repository under: `data_processing/`, and the condensed data can be found under `data_processing/data`.

3. Web Application

To enhance the usability and user experience, we developed a web application using Django as a client front for the MySQL database. The Check Your Plate web application is built with Django web application framework in combination with TypeScript, for managing front-end elements, SQLite for data storage. Visitors can view all existing recipes, but must register or login to enable the create recipe function. At the current build, the check your plate web app can provide users with the following services:

1. Any visitor to the website can view the details of existing recipes, including their detailed ingredient list and its nutritional content
2. A visitor to the site can register to become a member of the website.
3. Registered users can access their user information and modify them at any given time.
4. Registered users can create new recipes and modify existing recipes.

Future Work:

At the current stage of the Check Your Plate project, due to human error and technical issues such as cross-OS compatibility, the aforementioned MySQL database was actually not integrated with the Django web application, which runs on a SQLite database. As the web application has been developed quite extensively, the SQLite database cannot be easily swapped to MySQL database.

One obvious next course of action is to find a way to integrate the MySQL database with the web app. Two approaches are drafted at this point: The first is to carefully modulate the current

version of the Django web application, and carefully peel off the SQLite database that's connected into individual modules, replacing it with the MySQL database, and finally integrating all the modules back into one. This approach requires detailed planning, delicate version control, and a significant amount of time. The second proposed solution is to build another web front anew, using the MySQL database from the start. As for why preserving the MySQL backend over the web application, it is because the MySQL database actually contains better processed data, more carefully designed schema, and ready-to-use stored procedures. Not to mention that MySQL simply has higher storage capacity than SQLite, which is meant to be a lightweight database for non-data intensive web application use cases. So for scalability and current status of content, preserving MySQL databases is the optimal solution.

After fully integrating the two parts of the project, we would like to further build Check Your Plate so that it can achieve its initial goal of being a customizable diet reference guide. This requires data of Daily Reference Intakes for populations from different genders, age groups, and activity levels. Thus finding reliable technical data sources is an absolute must for scaling Check Your Plate to the next phase.

Project Contribution:

Yujia Cheng was responsible for the data processing, designing, implementation, testing and maintenance of Check Your Plate MySQL database.

Jiayi Zhang was responsible for the development of the Check Your Plate Django Web application.