SQL 6



Team Project Final Submission

INST 327-ESG1: Database Design and Modeling

May 10, 2019

# 



## The SQL6

Ebonie Gadson

Leo Giovanini

Eugenia Lee

Mu’tassim Muhib

Nelson Reyes

Kalood Younis

# 

# Introduction

The original goal of the database was to *organize recipes from Munchies by chef*. The sample data is available on [www.munchies.vice.com](http://www.munchies.vice.com), a news/media outlet. Each recipe from the sample data contains information such as food preparation time, food cook time, picture, ingredients, and on some occasions a how-to video.   
 One late evening, a team member searched the Munchies website for a recipe. The team member forgot the name of the recipe but remembered the name of the chef. Many websites use search bars to facilitate the activity of searching so the team member thought the search would take a few seconds. Unfortunately the website lacked a search bar. Even worse, the recipes were organized in chronological order so that over 500 recipes were distributed among 129 webpages. The search process took longer than 30 minutes, and the team member knew that reorganization of the information could optimize the search process. The team member thought to himself, “Why wasn’t the information searchable by chef or ingredient?” The goal of the project arose from an everyday problem.

The database is made of eight tables. The eight tables are: recipes, recipe\_tags, website, chef, tags, media, measurements and ingredients. Two important features of the design must be highlighted. Firstly, one recipe can have multiple media (images, links, etc), recipe tags, and measurements. Secondly, recipes, recipe tags, and measurements are linking tables for more specific tables like tags, ingredients, etc (as suggested when structuring for third normal form). An entity relationship diagram, included in the section titled “Logical Design”, outlines the organization of information. The ERD identifies the properties of each entity in each table as primary key, foreign key, non-key, one-to-one relationship, one-to-many relationship, etc.

# 

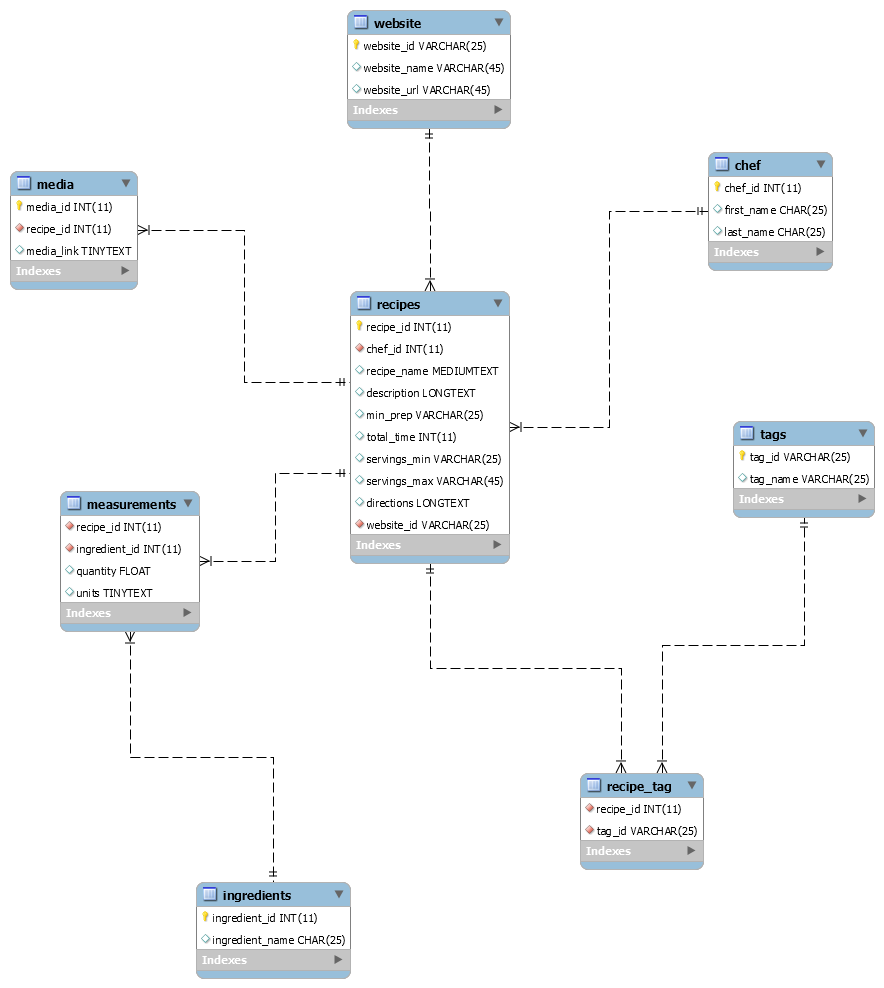
# Database Description

# Logical Design

When we created our model, we were a little confused about what attributes to include in it. We had scrapped our original dataset and wound up finding a concept that was more interesting. However, we were still unsure how we wanted to organize our theoretical data. One of us found a new source of data to scrape that was centered around the recipes. So, we made the recipes the main table to join everything together. There were some conflicts about how we should show each recipe because there are many ingredients involved, and many possible tags. This was resolved by putting each in a respective table, and then put a joining table between the recipes and these catalog tables. We deleted the twitter values from both the chefs and websites tables, as well as the website\_id link between the two since the last report, as suggested.

Our main table is the recipes table. The recipes table includes data such as the recipe\_id, the chef\_id, recipe\_name, description of the recipe, minimum preparation time, total time, serving minimum, serving maximum, directions, and website id. We used the recipes table as the root from which to branch out other tables. The recipe table also links with the recipe tag table which contains the recipe ID and a tag id. The recipe tag table links to the tags table which contains the tag id and the tag name. The media table contains the media ID, recipe id and the media link of the recipe. The website table contains the website id, website name, and the website url. The measurements table which contains the associated recipe ID, the associated ingredient ID, the quantity, and lastly the units of measurement. The measurement table links to the ingredients table which contains the ingredient ID and the ingredient name. The last table is the chef’s table which contains the chef ID, the chefs first name and the chef last name.

Figure 1: Food Database ERD



Description: Eight tables. Table names include chef, recipes, website, media, measurements, ingredients, recipe tag, tags.

# Physical Design

Creating the database was much easier once the structure of the entity relationship diagram was settled on and we created a model. We forward engineered the model and began populating it with data once we had confirmed that it was normalized. The team split the work to fill most of the tables first, then reconvened to share data and decide how to do the remaining work. Some of us wrote queries and tested the views while others finished inputting data into the last few tables. After all of that was finished, we shared files again, created a backup database, and confirmed that our structure, data, and views were intact when we ran the backup to recreate the database.

## Sample Data The sample data is from Github, the open source code exchange. The data was in JSON format, so we wrote a Python script that manipulated the data into a readable format. The JSON file provided us information such as the name, source, preparation time, wait time, cooktime, serving size, comments, amount of calories, amount of fat, saturated fat, fiber, carbs, sugar, protein, and it provides instructions. The scope of the project is larger than the sample data. The JSON file lacked media links, tags, and other important information. As a result, we created some of the sample data from our knowledge of food recipes. The sample data aided in the accomplishment of our goal because the sample data created a realistic prototype of the broad scope of the source.

# Views/Queries

“What recipes exist in the database? What should I prepare? Who created it?”  
Query 1 (whatCanIMake): The chef’s name, recipe name, ingredients, and quantities of every recipe. Recipes are sorted by recipe name, and then the chef’s last name.

“What can I make with noodles and what will I need?”  
Query 2 (sendNoods): The chef’s name, recipe name, ingredients, and quantities of recipes with noodles as an ingredient.

“My last name is Albanna! Are there any recipes from people with my last name?”  
Query 3 (theAlbannaBrothers): The chef’s name and recipes name by chefs with Albanna as the last name.

“I know an Alison. I hate her guts. Are there any recipes that are not associated with the first name ‘Alison’?”

Query 4 (numberOfIngredients): The chef’s name, recipes name, and count of ingredients of recipes by chefs without Alison as the first name.

“What noodle recipes are there from the Food Network?”

Query 5 (soups\_from\_foodnetwork): The recipe id, recipes name, and media link of recipes from foodnetwork with soup as a tag.

Figure 2: Queries Requirements Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| View Name | Req. A | Req. B | Req. C | Req. D | Req. E |
| whatCanIMake | x |  |  | x |  |
| sendNoods | x | x |  | x |  |
| theAlbannaBrothers | x | x | x | x |  |
| numberOfIngredients | x | x | x | x |  |
| soups\_from\_foodnetwork | x | x | x | x | x |

Description: The name of the query is on the left. The requirement is on the right. An “x” indicates that the query completes the requirements.

# 

# Changes from Original Design & Lessons Learned

The team designed a database for the information. In the process, the team realized that recipes existed on other websites. Taking a step back and seeing broad view of the scope, potential for organizing recipe information from many websites into one database existed. As a result, the team changed the goal of the project and redesigned the database. The data in the database come from a Github dataset. The broad scope increases the magnitude of improvement in searchability when queries are imagined are extremely large scale. A person using the database could search thousands of recipes by a view vivid memories like the main ingredient, part of the recipe name, a tag, or the source. Our impression was that reorganization of the information could improve browsing tasks and searching tasks.

A moment of crisis happened during data collection. That moment was the biggest challenge of the project. The sample data was broad enough so that recipes came from more than a singular source. Unfortunately, the sample data lacked media links and tags. At this point, we invested time into writing a Python script, cleaning the data, and importing the data. We did not want to exclude all data that the team member collected, so we created data that was a realistic enough simulation to fill the gaps.As a team, we discussed ideas in-person and online. We learned the importance of creating an open space and constructive atmosphere for discussion. We consistently met up in person after class and made it a point to voice our concerns through Slack and other means of direct messaging. We as a collective, and as individuals also voiced our concerns to our professor. Time management was also critical to our success. We began every meeting with a status update and ended the meeting with a plan of action.

While we value feedback about possible improvements for database, any suggestion that did not impact our grade was marked as “Potential Future Work.” Time was not on our side, so our top priority was creating a realistic enough simulation considering the scope of the project. The scope of the project changed halfway to the deadline and we adjusted the design so that time was not wasted.  
 We learned a lot of skills from this project. One skills is database design. We created an ERD for a database, and then reverse-engineered the code for the database. Another important skill is database administration. We imported data, imported/exported a database, created views, and created a backup database.

# 

# Potential Future Work

As it stands, the database shows that reorganization of the information improves searchability, and the database completes the project goal. A large area of potential future work exists as the database could be part of a recipe recommendation system. The recommendation system would consist of databases of dietary plans, food prices, and nutrition facts. In the near future, the recommendation system could improve the quality of life for many at-home cooks and professional chefs. A system could provide a user with a recommendation based on the user’s health information, financial information, and flavor pallet.

A small project for the future could scrape the data off the Munchies website, clean the data in Excel, and import the data into the current version of the database. The project could be tested as a prototype, so that we gain ideas from potential users about the views that a user might want or need.