Daily Recipe DB Model & DB Design Report

CS157A Introduction to Database Management Systems

San Jose State University

Team 5

Jiaxiang Guo

Soyeon Wang

Gricelda Tecun

Table of Contents

1. Functional Requirements

Functions

Login in / Sign Up:

- Users will be required to sign up for their own accounts to get access to the website.
- Users will be able to login to their own account to visit the website.

Search for Recipes:

- Users will be able to search for a recipe using the title as a criterion. The search functionality will provide the users with a shortcut to find their targeted recipes if posted before by the users.

View Profile:

- Users will be able to view their profile that their information of username, age, and country are displayed.

Edit Profile:

- Users will be able to edit their profile, such as editing the username and other information. Also, a changing password option is provided in this function.

View Recipe:

- Users will be able to see the recipes posted by other users. In other words, the user will be able to access the Recipe Forum, where other user's recipes are posted and ordered by date of creation.

Create Recipe:

- User can create a new recipe with a list of ingredients, time of preparation, photos, personal comments about your new recipe, process of preparation,
- User can create their own recipe with the elements they want to add, but the most important will be the name of the recipe, the ingredients, and process of preparation

Subscribe / Unsubscribe:

- Users will be able to subscribe another user who uploaded their favorite recipe, so they can follow up on his/her newest updated recipe. Also, an unsubscribe option is provided for users to unfollow whom they are no more interested in.

Report an inaccurate recipe:

- Users will be able to report inappropriate recipes that have cross contamination or that are harmful for human consumption. We need to make humans aware of what they are eating because there are many animals, plants, and other ingredients that are not suitable for intake.
- If some users keep reporting without proper reasons, users would be warned and cannot use the site for a certain period of time.

Bookmark Recipe:

- Users will be able to add the recipes they liked to a bookmark list.
- Users will be able to have a list of bookmarked recipes, so they can just look at their list quickly.
- Users will be able to delete a recipe from their bookmark list.

View My Favorite Recipes:

- Users will be able to access the recipes that they added in their favorite list.

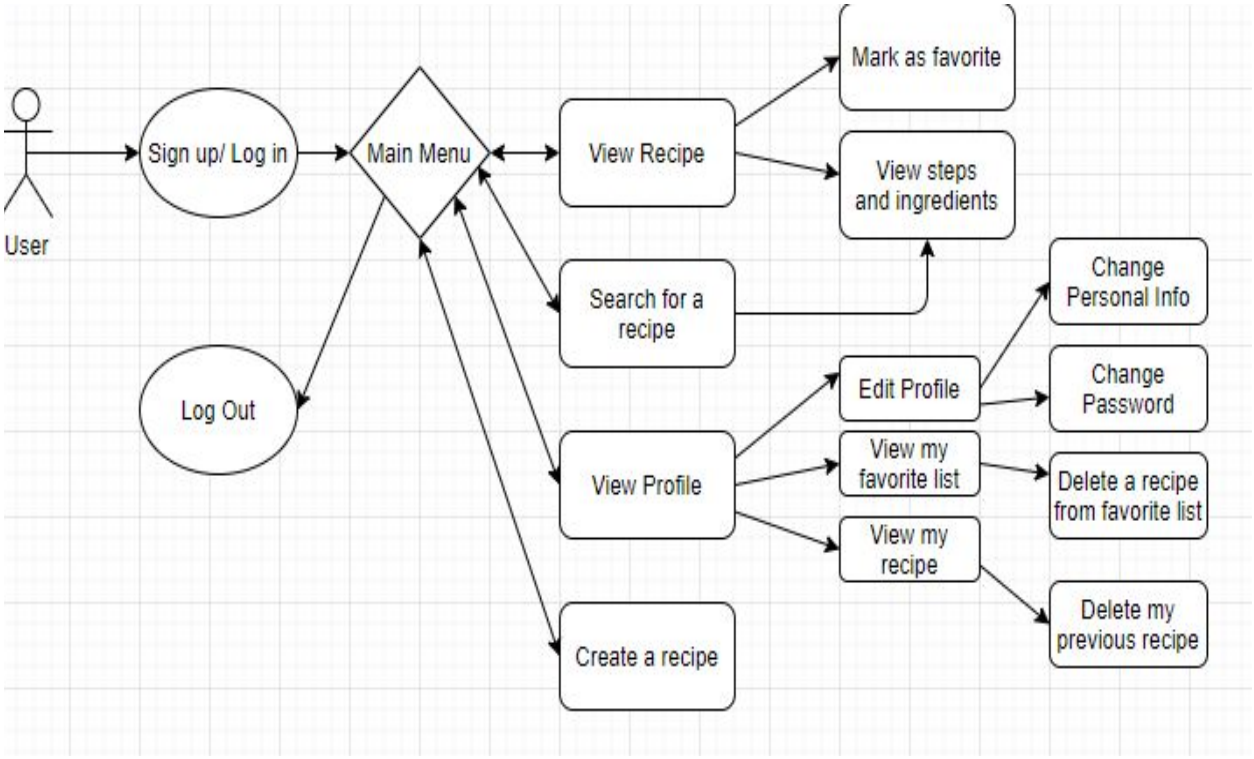
Delete Recipes:

- Users will be able to delete their own recipes that they already posted.

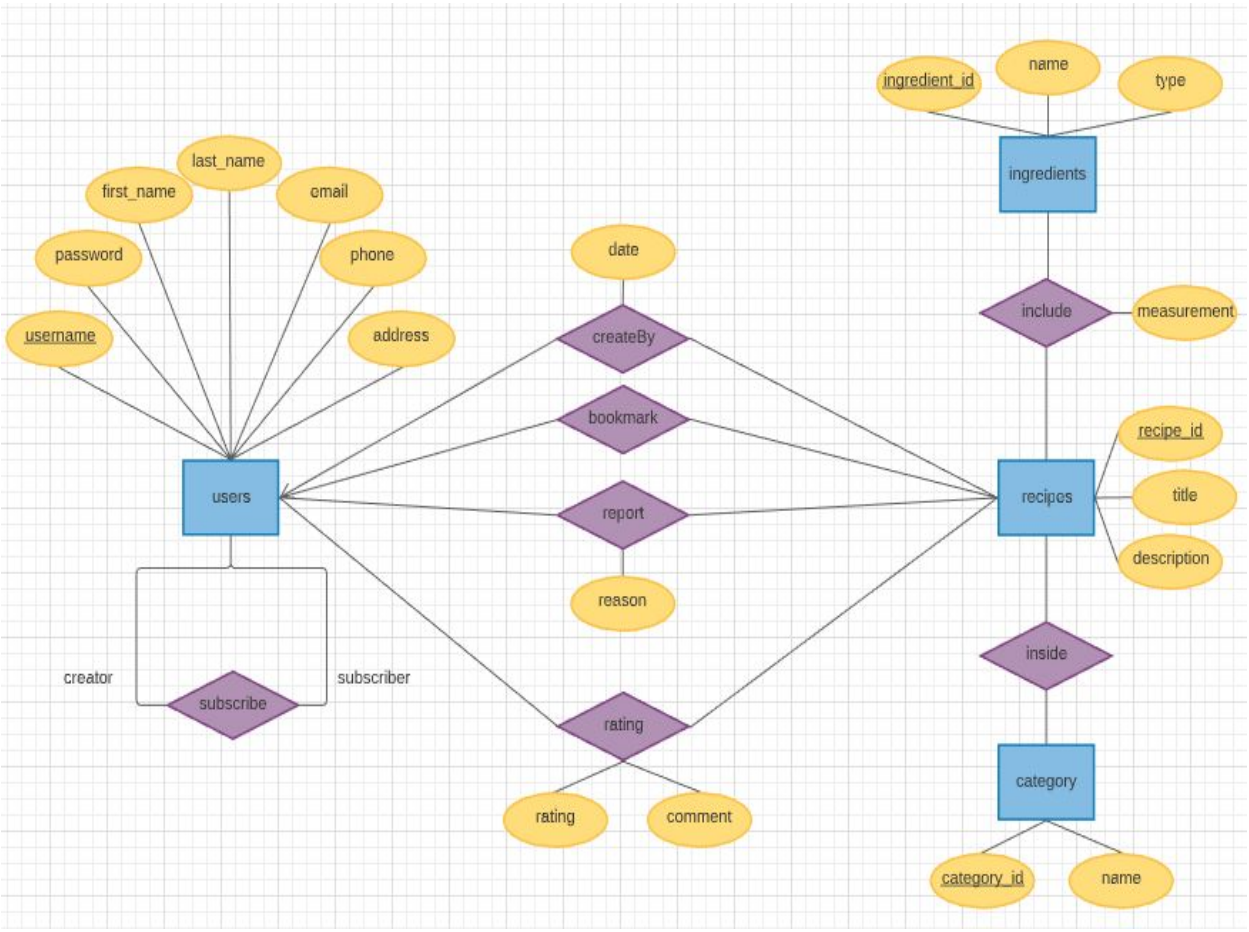
Share the link:

- Users can share the link of their recipes on social media such as Facebook, Instagram, etc.

User Flow Diagram



2. Entity Relationship Diagram



3. Relational Schemas

- **Entity-set schemas**

  users(<u>username</u>, password, last_name, first_name, email, address, phone)

  - User table contains every user login information.

  recipes(<u>recipe_id,</u> title, description)

  - Recipes table contains every recipe information created by the user.

  ingredients(<u>ingredient_id</u>, name, type)

  - Ingredients table contains every ingredient that a recipe includes.

  category(<u>category_id</u>, name)

  - Category table contains types where a recipe belongs to.

- **Relationship-set schema**

  subscribe(<u>creator_username</u>, <u>subscriber_username</u>)

  - Subscribe table contains user's favorite recipe uploader which is a user as well. Self-relationship with users table.

  createdby(<u>username</u>, <u>recipe_id</u>, date)

  - Createdby table contains a relationship between users table and recipes table that shows who is the creator of the recipe.

  bookmarks(<u>username,</u> <u>recipe_id</u>)

  - Bookmarks table contains a user's favorite recipe, it shows the relationship with users table and recipes table.

  report(<u>username</u>, <u>recipe_id</u>, reason)

  - Report table contains a reported recipe and the reason why it is reported. it is the relational table between uses table and recipes table.

  include(<u>recipe_id</u>, <u>ingredient_id</u>, measurement)

- Include table contains all ingredients that are needed for a recipe. It is the relational table between recipes table and ingredient table.

rating(<u>username</u>, <u>recipe_id</u>, rating, comment)

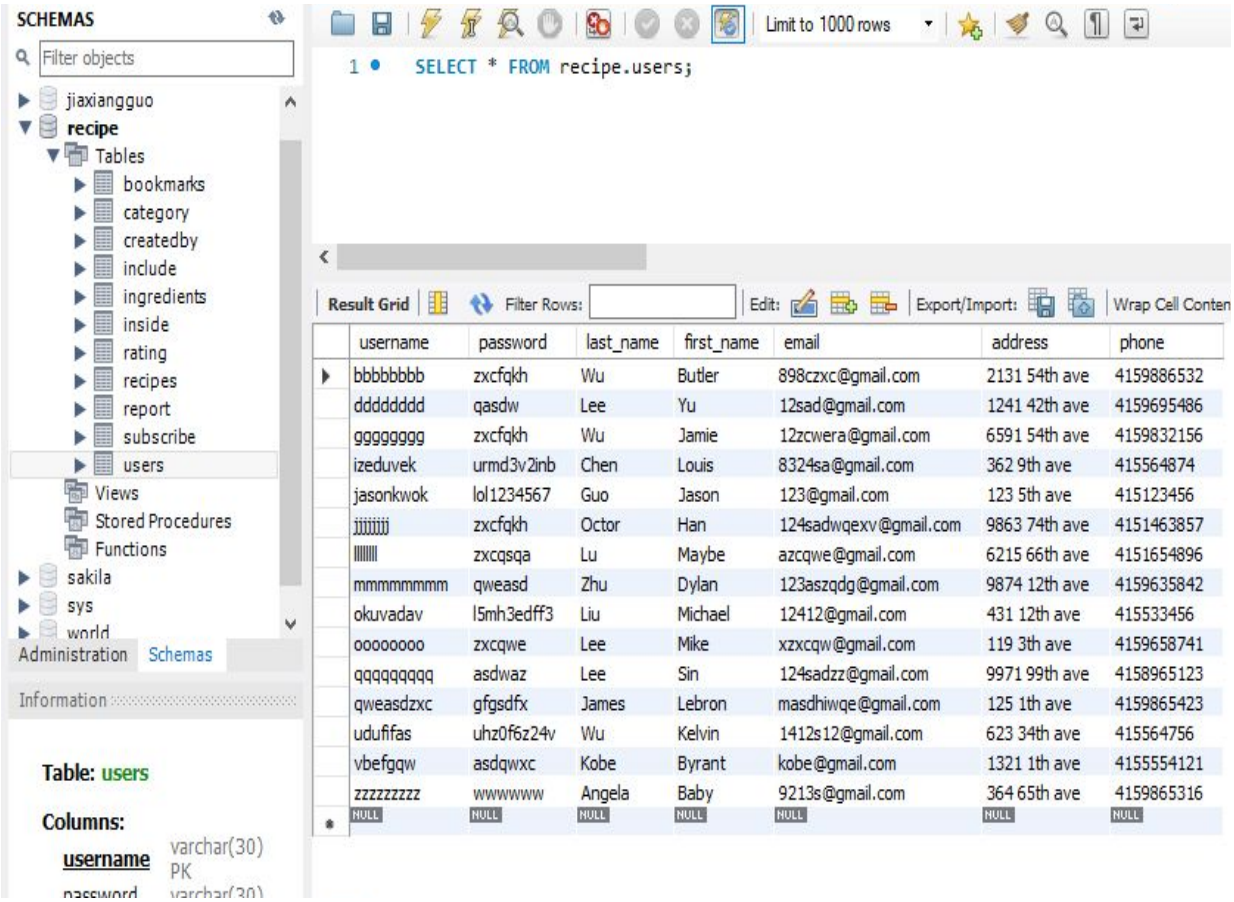
- Rating table contains all rating scores from users for recipes. It is the relational table between recipes table and users table.

inside(<u>recipe_id</u>, <u>category_id</u>)

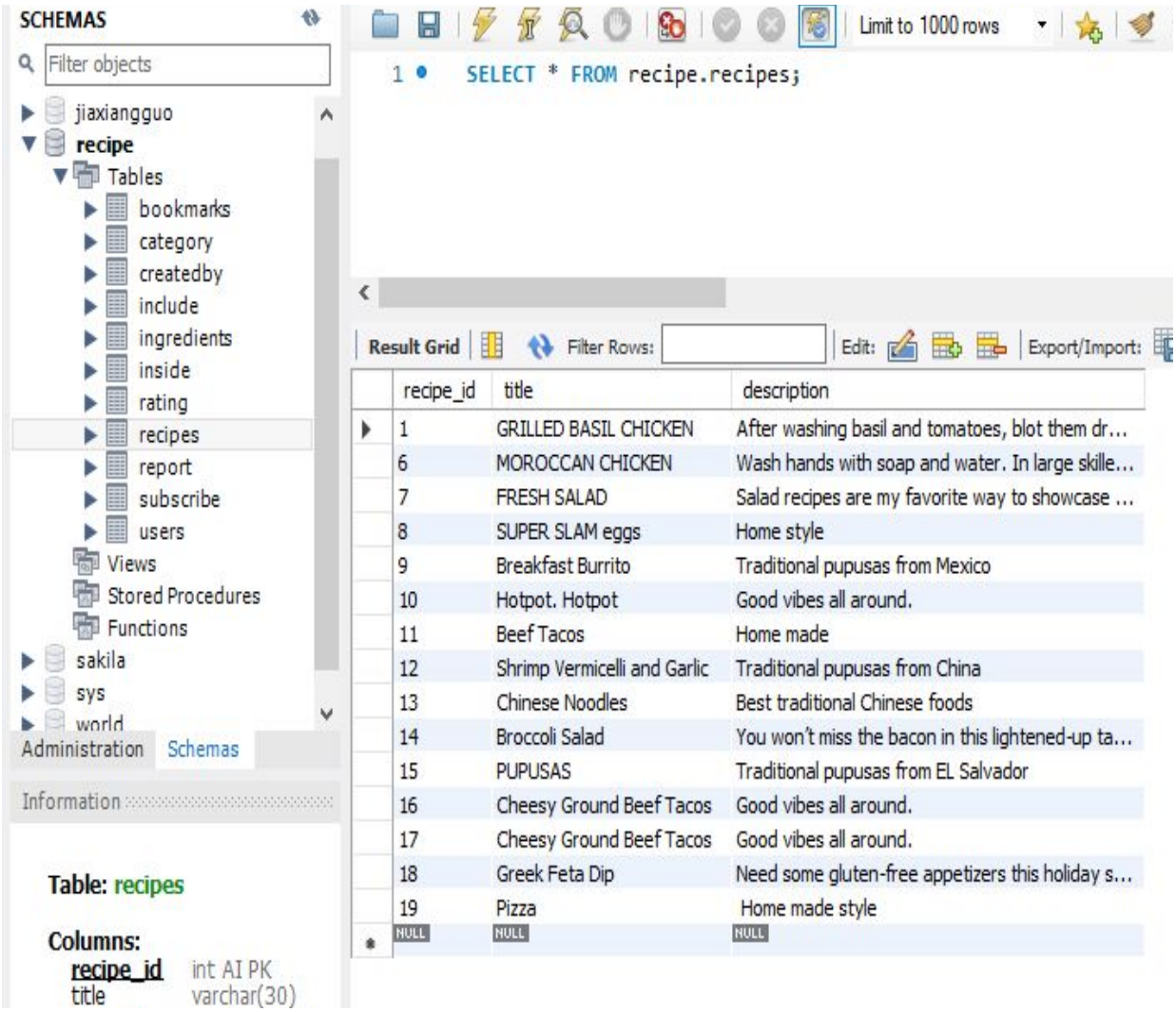
- Inside table contains all the recipe's categories. It shows where a recipe should be in. It is the relational table between recipes table and category table.

4. Screenshot of Tables

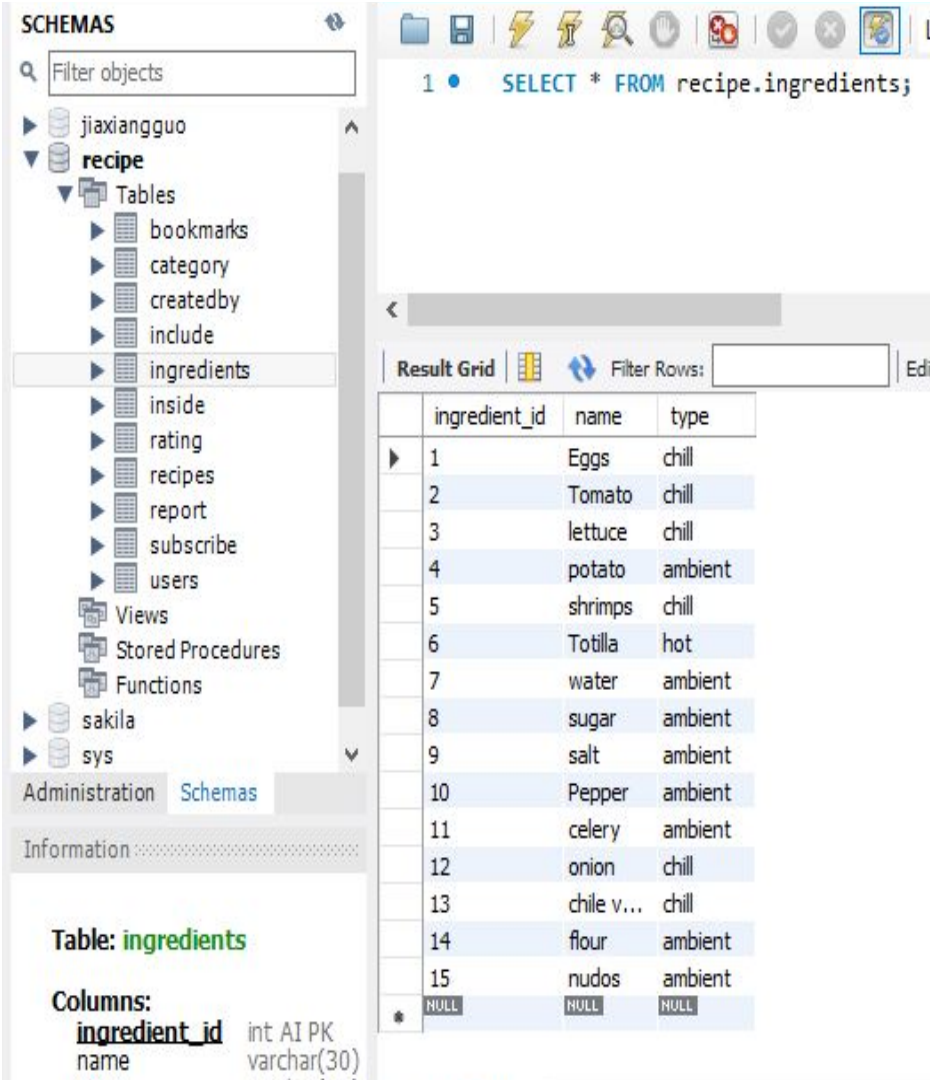
- users(<u>username</u>, password, last_name, first_name, email, address, phone)



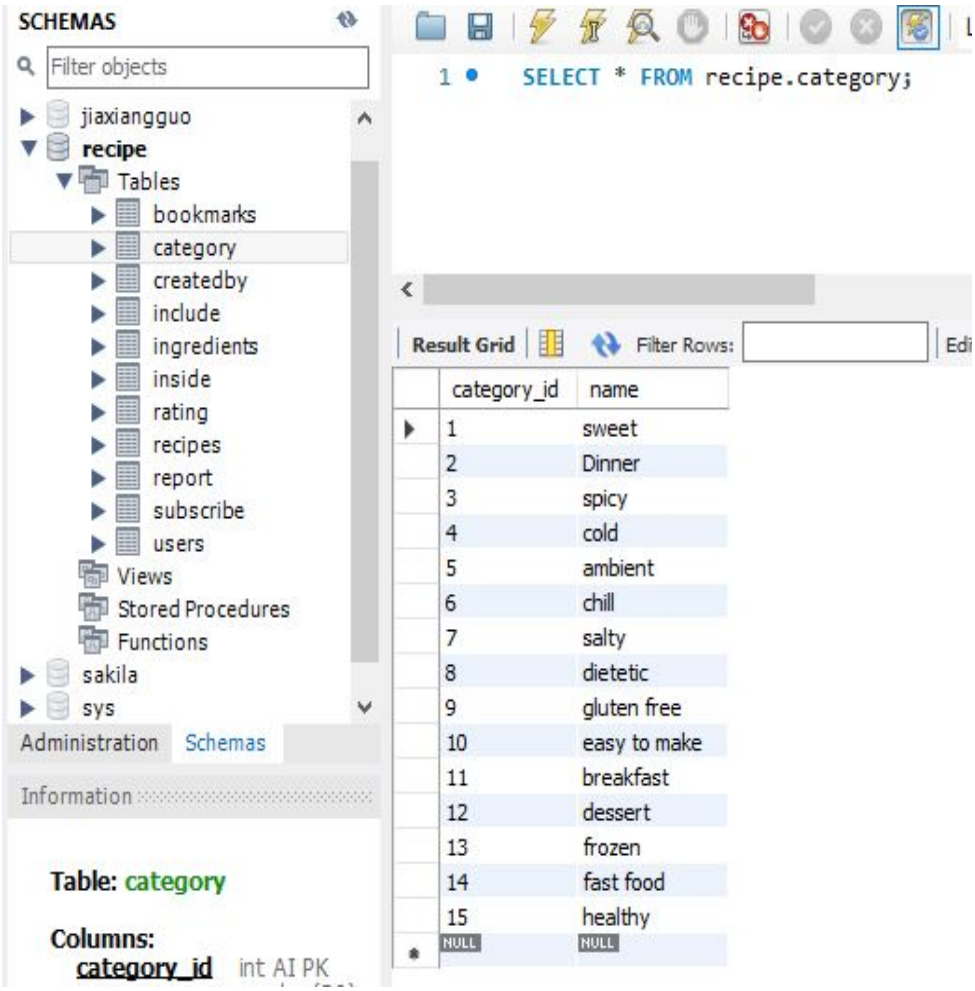
- recipes(recipe_id, title, description)



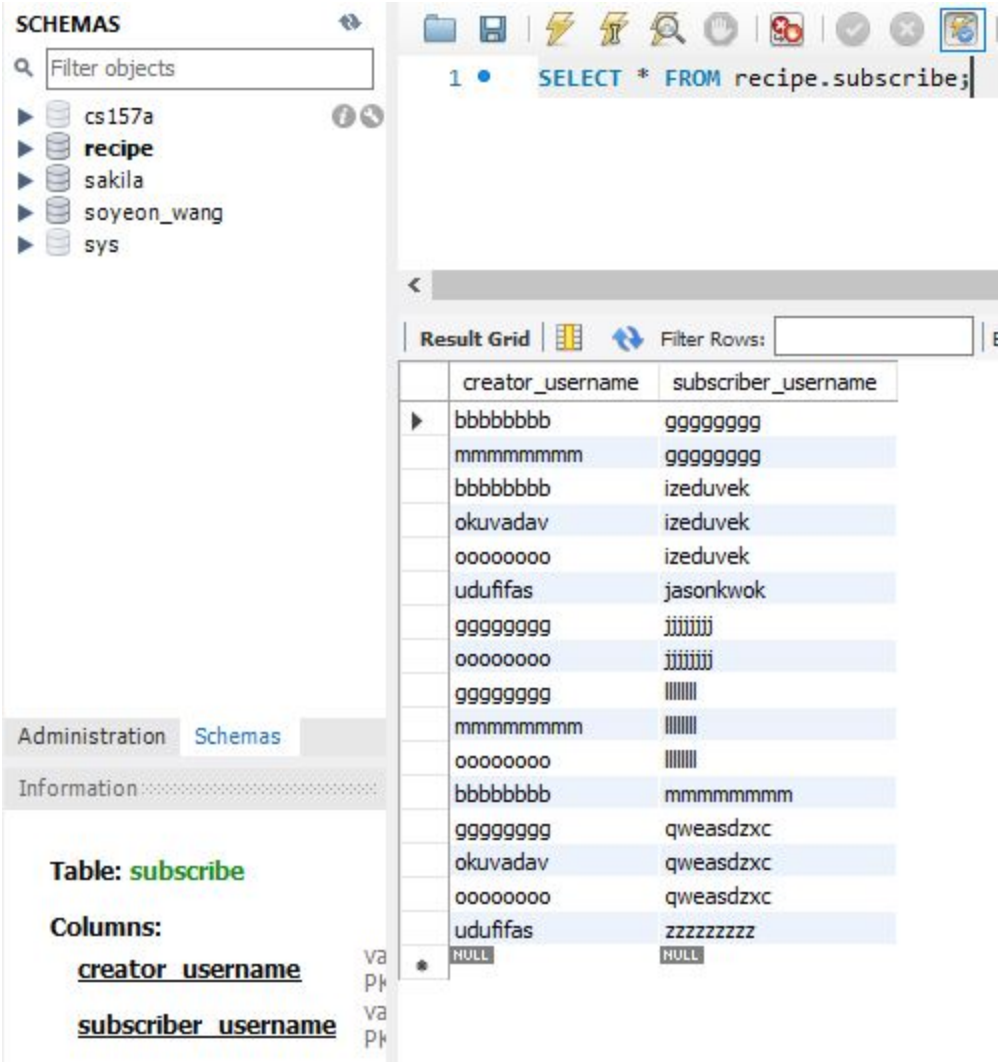
- ingredients(<u>ingredient_id</u>, name, type)



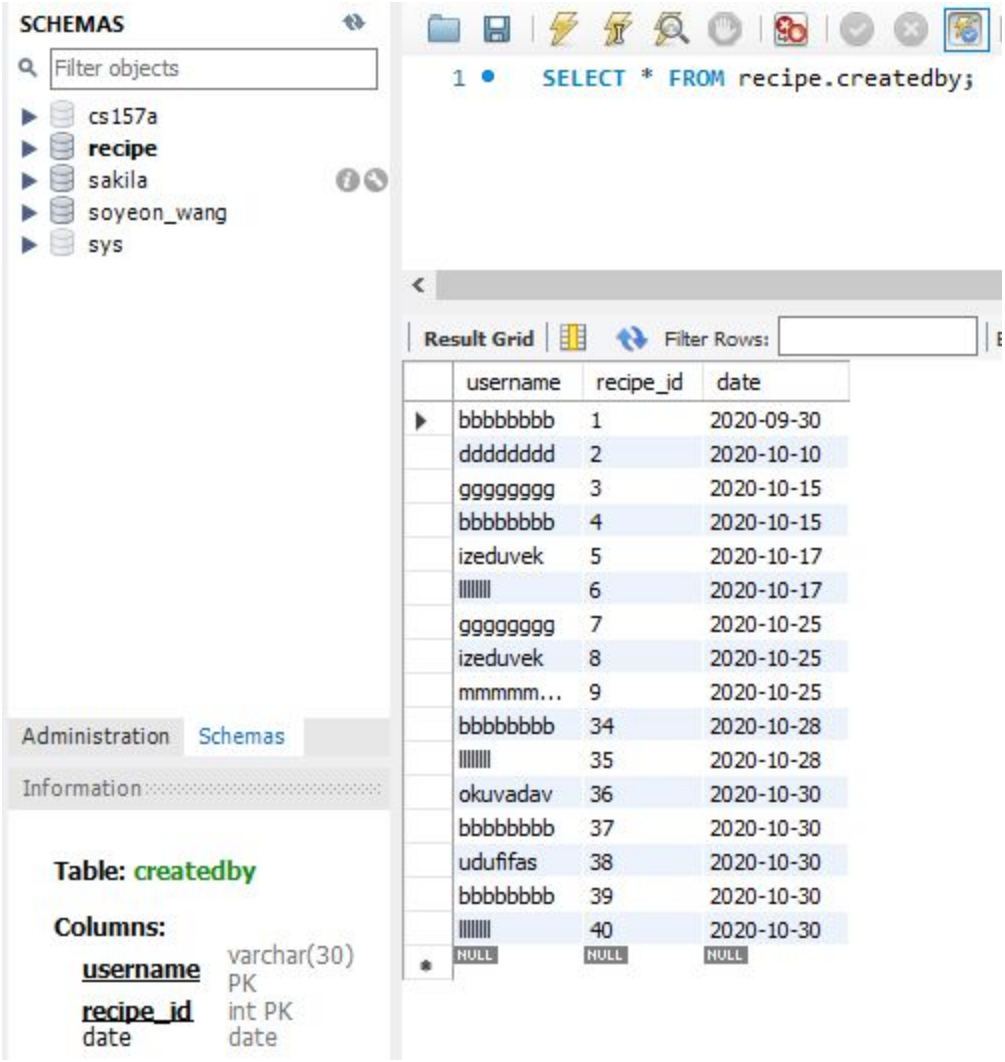
- category(<u>category_id</u>, name)



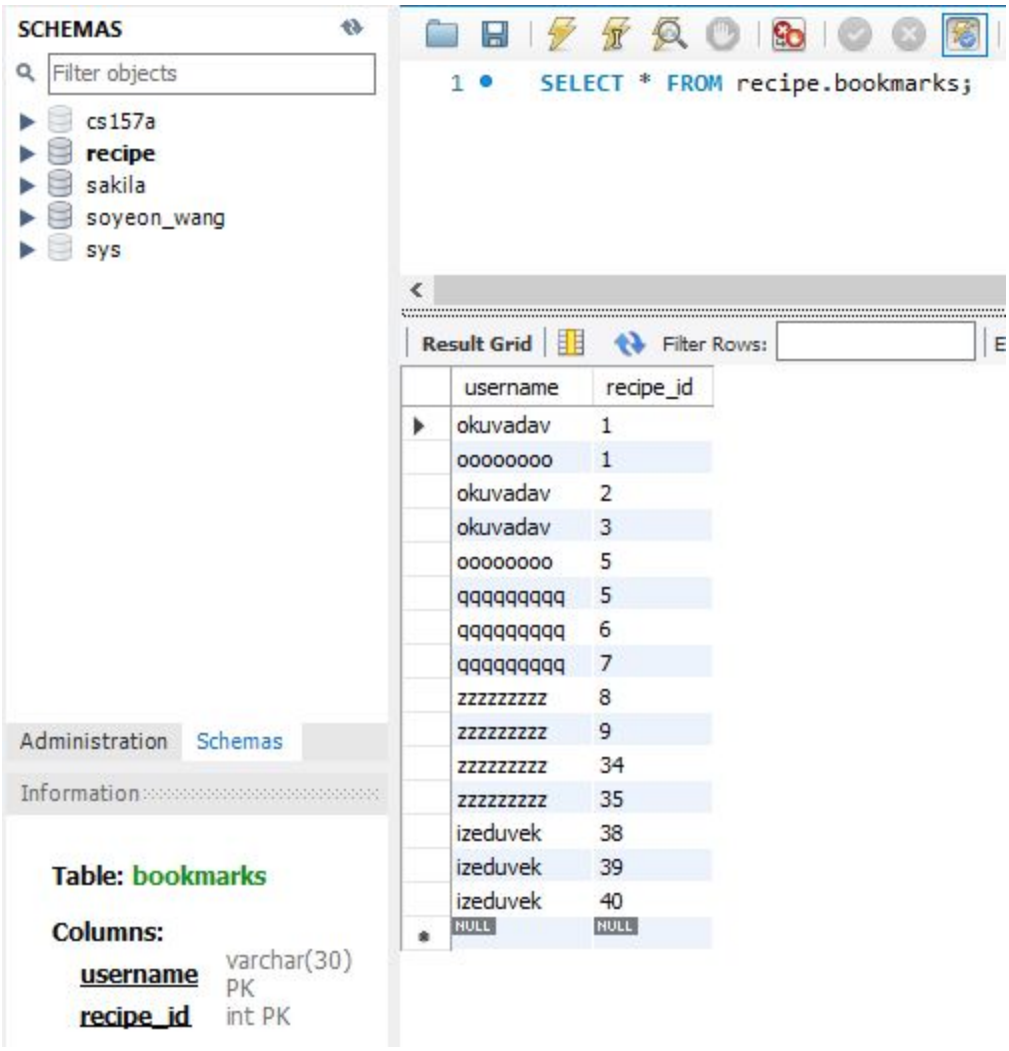
- subscribe(creator_username, subscriber_username)



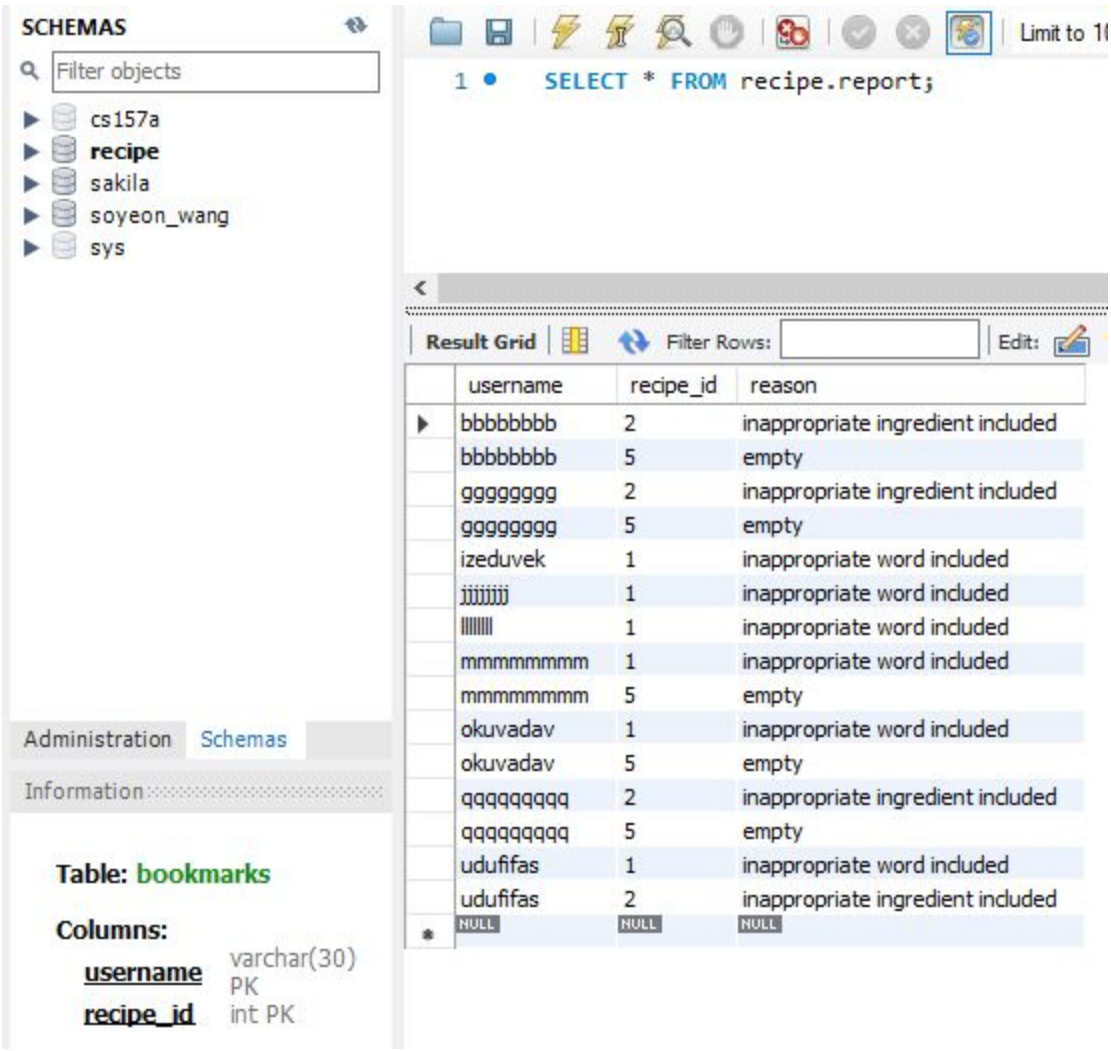
- createdby(<u>username</u>, <u>recipe_id</u>, date)



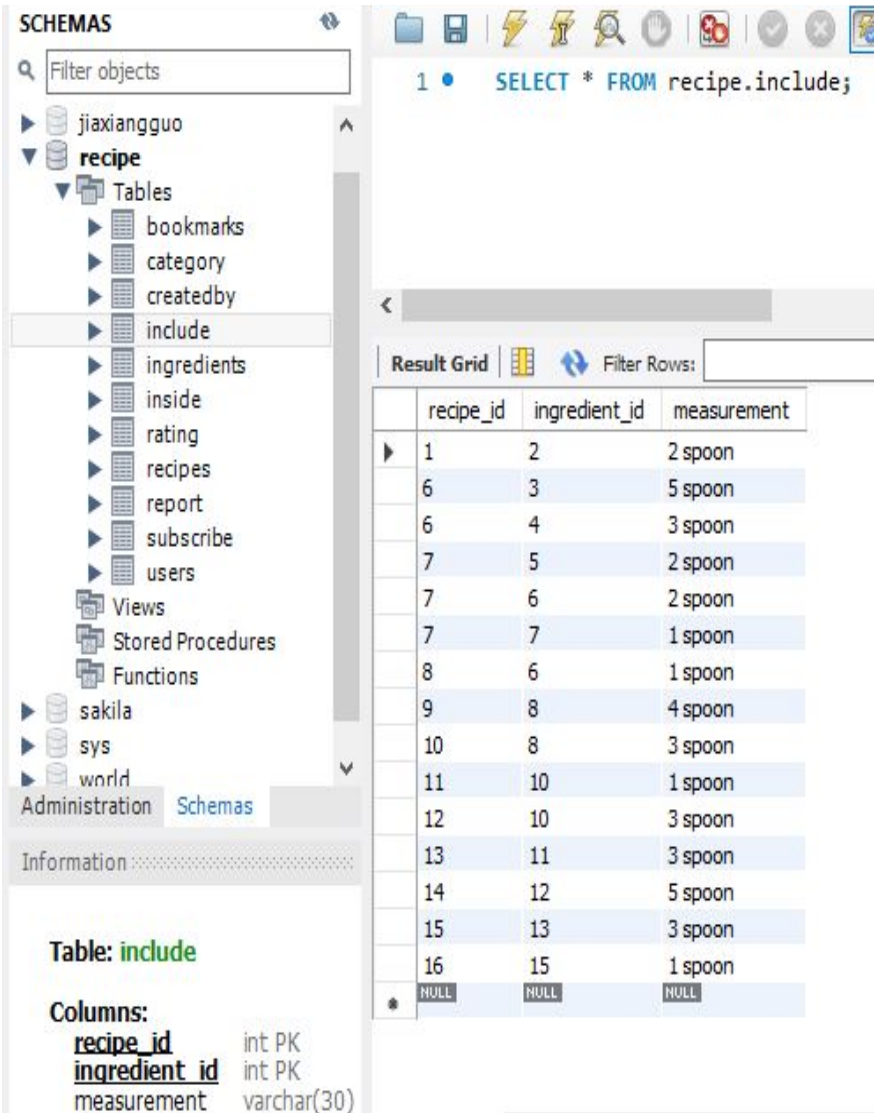
- bookmarks(<u>username</u>, <u>recipe_id</u>)



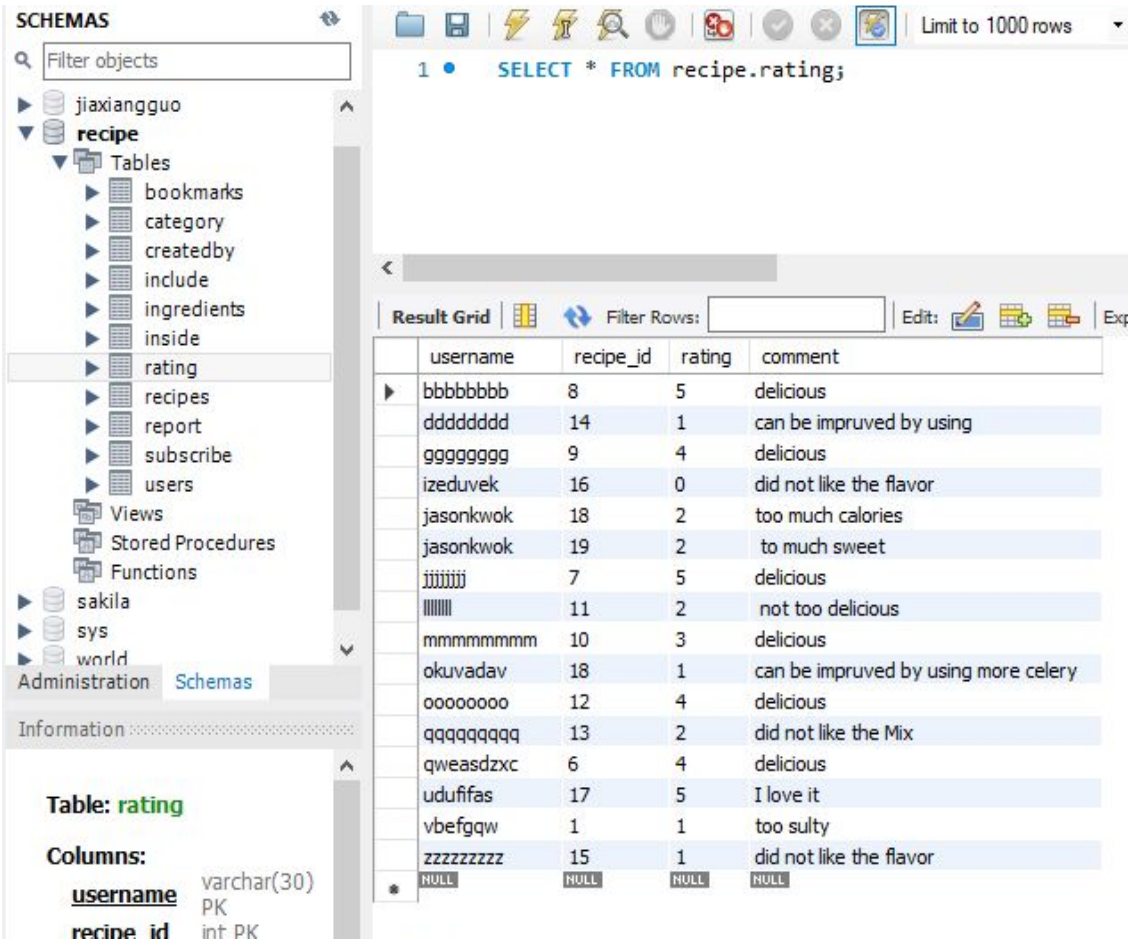
- report(<u>username</u>, <u>recipe_id</u>, reason)



- include(recipe_id, ingredient_id, measurement)



- rating(<u>username</u>, <u>recipe_id</u>, rating, comment)



- inside(recipe_id, category_id)