Gabriel Ogungbade **Food Recipes Database: Project**

2214105 **Scope**

Muh-Bashir Salami

2212366

Databases 2024 4/10/2024

ITMI23SP

Course Lecturer:

Reijonen Miika

**INTRODUCTION**

The aim of the food recipes database design project is to design and develop a comprehensive database for managing food recipes; the data stored inside it will include – and is not exclusive to – recipes, ingredients, cooking hardware required for each recipe, instructions, sample images…

**PROJECT PLANNING**

The database schema was developed with the understanding that many-to-many relationships will be apparent in the design. The schema diagram contains 6 entity tables, namely, Recipes, Ingredients, Optional Ingredients, Instructions, Cooking Hardware, Images, Categories.

The primary keys of each entity table were identified, and the foreign keys were assigned with correlation to their relations to other entities.

**IMPLEMENTATION**

Following the schema, the database tables were created, and they were populated with sample data. In the process of implementing the database, some realisations were made which led to the updating of the schema diagram.

Key constraints were properly assigned. And the database has one junction table, which is used to lay the groundwork for the functionality where recipes can be searched based on the availability of ingredients – this also permits the system to be able to recommend recipes based on the same parameters.

To build upon the requirement that users are given the privilege to make changes to the recipe database (create new recipe, update, or delete existing recipes), a CHECK constraint was implemented which requires the user’s role to either be ‘admin’ or ‘contributor’. This limitation is because of data integrity and security; some users might pose a threat to the system.

The database functions as it was intended to after its successful implementation, to demonstrate this some queries were written to test it.

**TABLE DESCRIPTIONS**

**RECIPES**

The Recipes table is a crucial component of the FoodRecipe database designed to store detailed information about various recipes featured in the application. Each entry into this table is unique to a specific Recipe.

**Uses**:

* The Recipes table acts as a central repository for all the recipes on the database.
* It facilitates search and retrieval to some extent.
* The Recipes table facilitates various ways for users to interact within the application, such as browsing recipes, viewing detailed descriptions, and checking ratings.
* With collected (and structured) data on cooking duration and user ratings, the table allows for analysis and reporting, such as identifying the most popular recipes or those with the highest ratings.­­ The ‘Ratings’ column uses a CHECK constraint that allows the users to rate using the values, 'good', 'bad', 'decent', 'excellent'.

**Relationships:**

There are two Foreign Keys in the Recipes table, namely CategoryID and UserID. The former is associated with the categories table, it helps to accurately place each recipe entry into a category or sub-category. The latter is linked to the Users table and serves to keep track of the creators of the recipes.

**INGREDIENTS**

The ‘Ingredients’ table is useful to keep track of the ingredients needed for different recipes. By having a dedicated table for ingredients, the application can ensure consistency and accuracy in ingredients across various recipes.

**Uses**:

* As stated above, the table provides a structured and centralized repository for all ingredients used in various recipes, ensuring consistency and accuracy in data collection.
* By storing ingredient data separately, the table allows for easy use of ingredients in multiple recipes without redundancy, streamlining recipe creation and management.
* The table aides in inventory management and procurement planning; it can be used to track the quantities of ingredients required for recipes.
* The table allows for detailed reporting and analysis of ingredient usage, which helps to identify frequently used ingredients and optimize ingredient purchasing and storage.

**Relationships:**

Through a many-to-many relationship, each ingredient can be associated with multiple recipes, and each recipe can include multiple ingredients. This relationship is managed with the junction table, RecipeIngredients.

**INSTRUCTIONS**

This table was created to manage and organize the procedural information of recipes efficiently. It allows for the detailed breakdown of the cooking process, ensuring that each step is documented and accessible. By having a dedicated table for instructions, the application can provide comprehensive guidance for users on how to prepare each recipe.

**Uses:**

* Detailed recipe guidelines and organized recipe steps.
* The use of ‘VARCHAR(MAX)’ for ‘InstructionDescription’ allows for instructions of any length, accommodating both simple and complex cooking steps without limitations.
* Through the ‘RecipeID’ foreign key, the table connects detailed instructions to specific recipes, maintaining a clear relationship between the recipe data and its preparation steps.
* With the Recipe Instructions being very detailed, the users can access detailed cooking steps directly form the application, improving their cooking experience and helping them achieve better results with each recipe.

**Relationships:**

The table is directly related to the Recipes table via the RecipeID foreign key. This relationship ensures that each set of instructions is linked to a specific recipe, allowing users to follow the steps required to prepare each dish.

**COOKINGHARDWARE**

This table was added to manage and organize the details of cooking hardware efficiently. It allows the application to provide users with a comprehensive list of equipment needed for each recipe. By having a dedicated table for cooking hardware, the application can maintain a clear and organized record of equipment requirements.

**Uses:**

* It provides a structured way to store and manage data regarding the hardware or equipment required for each recipe.
* The CookingHardware data is stored separately, this permits the users to select from the arsenal of equipment and use the cooking hardware in multiple recipes barring redundancy.

**Relationships:**

The ‘CookingHardware’ table does not contain any Foreign Keys. By omitting explicit relationships, the table maintains a simplified structure, reducing complexity and potential overhead. It also provides flexibility in how the association between hardware items and recipes are implemented. It also allows each hardware to be used in various recipes.

**IMAGES**

This table was created to enrich the user experience by providing visual representations of recipes. Images can help users better understand a recipe, visualize the result, and make informed decisions about which dishes to prepare.

**Uses:**

* The table enhances the user experience by providing visual representations of each dish.
* Images complement recipe descriptions and instructions, providing users with additional context and visual guidance.
* Images stores in the table can be used for marketing and promotional purposes.

**Relationships:**

The Images table has one Foreign Key, ‘RecipeID’, each entry into the table corresponds to a specific recipe in the ‘Recipes’ table. This relationship allows for the association of one or more images with each recipe.

**CATEGORIES**

This table was added to organize and classify recipes based on their respective categories or types. It provides a structured way to group similar recipes into categories or sub-categories, making it easier for users to browse and discover recipes of interest.

**Uses:**

* As mentioned above, it provides a structured way to organize and classify recipes into different categories or types based on common attributes such as cuisine, meal type, dietary restrictions, cooking method etc.
* The table enhances browsing experience.
* Categories can be used as filters or search criteria to refine recipe search results.
* Users can customize their browsing experience by selecting favourite categories or excluding categories they are not interested in.
* Categories help maintain a structured organisation of recipe content within the application, this makes it easier for administrators to manage and update recipes.

**Relationships:**

The ‘Categories’ table is standalone, i.e. it has no foreign keys. Although it is referenced in the Recipes table, and each recipe entry has a ‘CategoryID’ stored within it.

**RECIPEINGREDIENTS**

The ‘RecipeIngredients’ table is designed to establish the relationship between recipes and their corresponding ingredients, detailing the quantity and measurement units of each ingredient required for a recipe.

This table was created to manage and organize the ingredient information for recipes within the application. It is a junction (or associative) table that connects recipes with their constituent ingredients, allowing for precise tracking of ingredient quantities and measurement units for each recipe.

**Uses:**

* It establishes a direct relationship between recipes and their constituent ingredients, allowing for precise mapping of which ingredients are used in each recipe.
* The table provides a structured format for storing and accessing ingredient details associated with each recipe.
* The table allows for easy modification of recipes by updating ingredient quantities or adding/removing ingredients. This flexibility enables recipe adjustments based on factors such as serving size, dietary preferences, or ingredient availability.
* The table also supports recipe search and filtering functionalities by enabling users to specify ingredient criteria.
* The table supports the functionality which permits the user to get alternative recommendations (or similar recipes) that belong to the same category.

**Relationships:**

The ‘RecipeIngredients’ table serves as a crucial intermediary entity within the database, establishing two primary relationships between the Recipes and Ingredients tables. Ultimately this is a many-to-many relationship: Since a recipe can include multiple ingredients, and an ingredient can be used in multiple recipes, this structure allows for versatile associations between recipes and ingredients. In other words, multiple recipes can share the same ingredient, and a single recipe can include multiple ingredients. This setup enables flexible and dynamic management of recipes and ingredients within the database, accommodating diverse culinary compositions and variations.

**USERS**

This table allows the application to distinguish between different users and customize their experiences based on their roles and preferences. Each entry into this table reperensents a unique user account. A check constraint has been implemented to ensure data integrity and security.

**Uses:**

* User authentication.
* User profile management: The table stores user profile information such as date of birth and email address, allowing users to maintain and update their personal details within the application.
* Access control: The ‘UserRole’ column defines the role of teach user within the application, distinguishing between administrators and contributors.
* User interaction tracking: While not explicitly stated in the table structure, the Users table can potentially be linked to other tables to track user interactions and activities within the application. This includes actions such as recipe creation, modification, or deletion, which can be associated with specific user accounts.

**Relationships:**

The ‘Users’ table does not directly reference or rely on other tables through foreign key constraints: It operates as a standalone entity within the database.

**FUNCTIONALITY TEST**

To demonstrate the database’s functionality, various queries, procedures and VIEWS have been created. The function of the queries and procedures are explained in the ‘functionality\_test\_queries’ document. Below is a brief description of the views provided in the functionality test.

**VIEWS**

**ShowAvailableRecipes:**

The ‘ShowAvailableRecipes’ view combines data from the ‘Recipes’ and ‘Images’ tables. It prints each recipe’s name, rating, total cooking time and image URL. A LEFT JOIN is used to include all the recipes (even those without images).

**RecipeInstructions:**

The ‘RecipeInstructions’ view combines data from the ‘Recipes’, ‘Ingredients’, ‘Instructions’, and ‘Images’ tables to provide a general overview of a recipe. It prints the recipe name, description, rating, ingredient details (name, quantity, unit), image URL, and cooking instructions (step number and description). The view uses INNER JOIN to link recipes with their ingredients and LEFT JOIN to include optional images and instructions.

**ADDITIONAL FEATURES**

An optional feature that could be added to enhance the food recipe management application is a user profile and personalized recommendation system.

With a user profile system, users can create accounts and save their favorite recipes, dietary preferences, allergies, and cooking equipment inventory. The system could then use this information to provide personalized recipe recommendations based on the user's preferences, available ingredients, and cooking hardware.

Additionally, the application could track user interactions such as recipe views, likes, and reviews to further refine and improve the recommendation algorithms over time.

This feature would enhance the user experience by tailoring recipe suggestions to individual tastes and preferences, making it easier for users to discover new recipes that they are likely to enjoy. It would also promote user engagement and retention by providing a more personalized and interactive experience.

**CONCLUSION**

The project successfully designed and implemented a comprehensive food recipe database. The database schema effectively manages various recipe components through six entity tables and leverages a junction table to enable recipe search and recommendation based on ingredient availability. Additionally, a security measure was implemented using a CHECK constraint to restrict data modification to authorized users.

While the core functionality is complete, future enhancements could include a user profile system and a personalized recommendation engine. This would allow users to save preferences, track dietary restrictions, and receive recipe suggestions tailored to their needs and available ingredients. Overall, the project delivers a robust foundation for a user-friendly food recipe management application.