

Agile Development Group Project Database Design Document

Author: Shane Waters [shw30]
Config Ref: CSM2020_WILSON_DBD
Date: 23rd February 2022
Version: 1.0
Status: Release

Department of Computer Science
Aberystwyth University
Aberystwyth
Ceredigion
SY23 3DB
Copyright © Aberystwyth University 2022

CONTENTS

CONTENTS	2
1. INTRODUCTION	3
1.1 Purpose of this Document	3
1.2 Scope.....	3
2. BODY OF DOCUMENT	3
2.1 Auth_user Table.....	3
2.2 Recipe Table	4
2.3 Ingredient Table	4
2.4 Nutrient Table	4
2.5 Database UML	5
REFERENCES	6
DOCUMENT HISTORY	7

1. INTRODUCTION

1.1 Purpose of this Document

This document describes and lays out an implementation of a normalised relational database for the project. This document provides an implementation of the database with example data.

1.2 Scope

Everything which is persistently stored for the project will be included within the document.

2. BODY OF DOCUMENT

The user table and password table are subject to change depending on Django' implementation of user login storage. The following tables are an initial design of the database to ensure the data follows at least 3rd Normal Form.

2.1 Auth_user Table

The user table is responsible for storing the user's login information. This table is essential for uniquely identifying a user within application and database. Some sensitive information is stored within this table, so it is required that it follows any legal requirements retaining to storing personal information as outlined by the Data Protection Act. Passwords are required to be hashes to ensure they are kept securely; this document will not cover the hashing of passwords.

- Id: PRIMARY KEY, INTEGER – Uniquely identifies a user, auto assigned by Django. Is used as the primary key for the auth_user table.
- Password: VARCHAR(128) – A hashed password.
- Last_login: TIMESTAMP WITH TIME ZONE – Records the last login time.
- Is_superuser: BOOLEAN – Determines whether the user has super user permissions.
- Username: VARCHAR(150) – The alias which the user is displayed by.
- First_name: VARCHAR(150) – The user's first name.
- Last_name: VARCHAR(150) – The user's last name.
- Email: VARCHAR(254) – The user's email address.
- Is_staff: BOOLEAN – Determines whether the user has staff permissions.
- Is_active: BOOLEAN – Determines whether the users account is active or not. It is preferred to make a user's account inactive rather than deleting it to maintain foreign key dependencies and maintaining their data.
- Date_joined: TIMESTAMP WITH TIME ZONE – The date which the users account was created and added into the database.

2.2 Recipe Table

The recipe table is responsible for storing the recipes which the user's create. This table holds most of the information for a recipe bar information that would otherwise stop the table conforming to 3rd Normal Form. The Ingredient and Nutrient table is where data absent from this table is located.

- Id: PRIMARY KEY, FOREIGN KEY, INTEGER - Uniquely identifies a user, auto assigned by Django. Is part of the composite primary key for the recipe table, is a foreign key from the auth_user table.
- Recipe_Name: PRIMARY KEY, VARCHAR(120) – The name of the recipe. Is part of the composite primary key for the recipe table.
- Cost: MONEY – How much the money the recipe will cost to make in GBP.
- Portion: SMALLINT – The number of portions the recipe will serve.
- Description: TEXT – A description of what the recipe is.
- Instructions: TEXT – A guide containing the steps to making the recipe.
- Cooking_time: SMALLINT – The amount of time the recipe will take to cook.
- Skill_level: ENUM("Easy", "Medium", "Difficult") – Categorises the recipe on how difficult it will be to make.
- Skill_level: ENUM("Party", "Dinner", "Picnic", "Desert", "Drink", "Other") – Categorises the recipe on how difficult it will be to make.
- Date_added: TIMESTAMP – When the recipe was created and added into the database.

2.3 Ingredient Table

The ingredient table is responsible for storing the ingredients used within a recipe. To keep the database in 3rd Normal Form all fields must be atomic, so each ingredient has its own entry into this table.

- Id: PRIMARY KEY, FOREIGN KEY, INTEGER - Uniquely identifies a user, auto assigned by Django. Is part of the composite primary key for the recipe table, is a foreign key from the auth_user table.
- Recipe_Name: PRIMARY KEY, FOREIGN KEY, VARCHAR(120) - The name of the recipe. Is part of the composite primary key for the recipe table, is a foreign key from the recipe table.
- Ingredient: PRIMARY KEY, VARCHAR(120) – The ingredient used in the recipe, this also includes the amount of the ingredient.

2.4 Nutrient Table

The nutrient table is responsible for storing all the nutrients of a recipe. This table has a one-to-one relationship with the recipe table and could be included inside it, it was separated out for clarity.

- Id: PRIMARY KEY, FOREIGN KEY, INTEGER - Uniquely identifies a user, auto assigned by Django. Is part of the composite primary key for the recipe table, is a foreign key from the auth_user table.
- Recipe_Name: PRIMARY KEY, FOREIGN KEY, VARCHAR(120) - The name of the recipe. Is part of the composite primary key for the recipe table, is a foreign key from the recipe table.
- Fat: PRIMARY KEY, DECIMAL(5, 2) – The amount of fat in the recipe.
- Carbohydrate: PRIMARY KEY, DECIMAL(5, 2) - The amount of carbohydrates in the recipe.
- Fibre: PRIMARY KEY, DECIMAL(5, 2) - The amount of fibre in the recipe.
- Protein: PRIMARY KEY, DECIMAL(5, 2) - The amount of protein in the recipe.
- Salt: PRIMARY KEY, DECIMAL(5, 2) - The amount of salt in the recipe.

2.5 Database UML

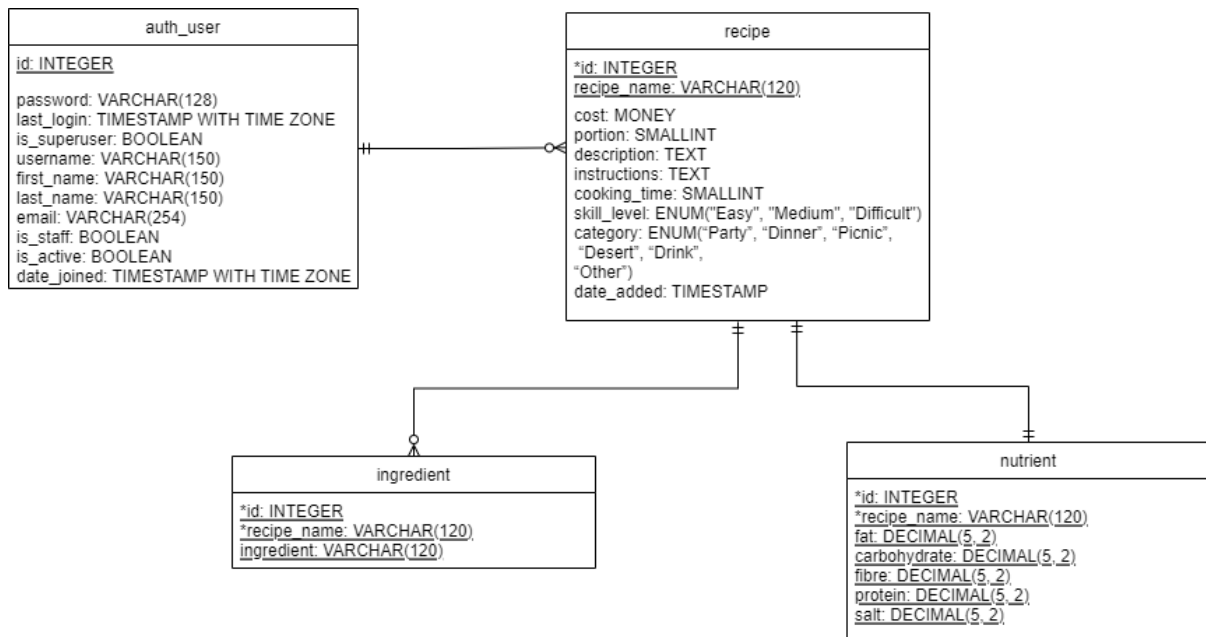


Figure 1: Database UML Diagram

REFERENCES

- [1] PostgreSQL Organisation, Datatypes, <https://www.postgresql.org/docs/9.5/datatype.html>

DOCUMENT HISTORY

<i>Version</i>	<i>Date</i>	<i>Changes made to document</i>	<i>Changed by</i>
0.0	20/02/22	Initial creation of the document	SHW30
0.1	23/02/22	Database tables and implementation design included and explained.	SHW30