

ВЫБОР БАЗЫ ДАННЫХ ДЛЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ COFFEE BREAK, ХРАНЯЩЕГО СПИСОК РЕЦЕПТОВ

*В.В. Видман, ассистент ОИТ,
И.А. Федотов, студент гр. 8К81
Томский политехнический университет
E-mail: iaf7@tpu.ru*

Введение

Сейчас существует бесчисленное количество разных рецептов кофейных напитков, а также у многих людей есть свой собственный, который нравится им больше всего. Каждый рецепт отличается от другого, это связано с различными ингредиентами, добавками, их пропорциях, кроме того, способ приготовления также может отличаться.

Суть моего приложения заключается в том, что пользователь может познакомиться с рецептами разных напитков или найти определенный, который его заинтересовал. А в дальнейшем проверить себя на знание рецептов, попробовав смешать все в нужных пропорциях и получив оценку соответствия полученного напитка. Также в приложении будет возможность добавить свой рецепт, который всегда можно будет легко найти и посмотреть.

Модель базы данных

Самым удобным способом хранения информации является база данных. Она позволяет легко найти и воспользоваться именно тем, что необходимо пользователю в данный момент, а также добавить новую информацию. Кроме того, правильно смоделированная база данных не хранит в себе ничего лишнего, информация представлена структурированно и оптимизированно, благодаря чему с ней легко работать в процессе разработки.

Для приложения необходимо хранить список рецептов различных напитков, каждый из них состоит из названия, способа приготовления и множества ингредиентов.

После анализа я получил две различных модели базы данных, каждая из которых имеет свои плюсы и минусы. Стоит отметить, что от модели сильно зависит процесс разработки приложения, поэтому важно выбрать наиболее подходящий вариант.

В первой модели вся информация хранится в одной таблице, каждая строка соответствует одному рецепту, а каждый столбец отвечает за определенный ингредиент, в итоге, на пересечении мы имеем количество определенного ингредиента для выбранного рецепта. Данная модель проста и наглядна, с ней легко взаимодействовать. Но в таком решении есть недостатки, каждый рецепт использует не все множество ингредиентов, поэтому многие ячейки будут пустыми или равны нулю, также пользователь может добавить новый рецепт, в котором будет уникальная добавка, для которой будет создан новый столбец.

Пример модели представлен в таблице 1.

Таблица 2. Первая модель базы данных

Название напитка	Способ приготовления	Кофе	Вода	Молоко	Сахар	Корица	Шоколад
Латте	...	30	250	200	20	2	0
Капучино	...	14	0	250	15	0	5
Эспрессо	...	7	30	0	7	0	0

Во второй модели вся информация разбита по частям и хранится отдельно, то есть одна таблица отвечает за количество кофе, другая за различные добавки и так далее. Преимущество данной модели в том, что не будет нулевых или пустых значений, если определённого ингредиента в напитке нет, то запись в соответствующей таблице будет отсутствовать. Также стоит обратить внимание на таблицы, отвечающие за сиропы, добавки и алкоголь, в отличии от первой модели, от увеличения количества ингредиентов, количество столбцов не меняется, просто для рецептов, где содержится данная добавка, добавляется строка с необходимой информацией. Еще одним преимуществом второй модели является удобство, при реализации некоторых функций, например, фильтр безалкогольных напитков, по первой

модели не получится так просто получить данную информацию, во второй же все легче, если запись в таблице с алкоголем об определенном напитке отсутствует, значит он безалкогольный. Но есть и недостаток, множество таблиц требуют связывания, что осложняет создание базы данных, но в нашем случае это незначительно.

Концептуальная структура второй модели представлена на рисунке 1.

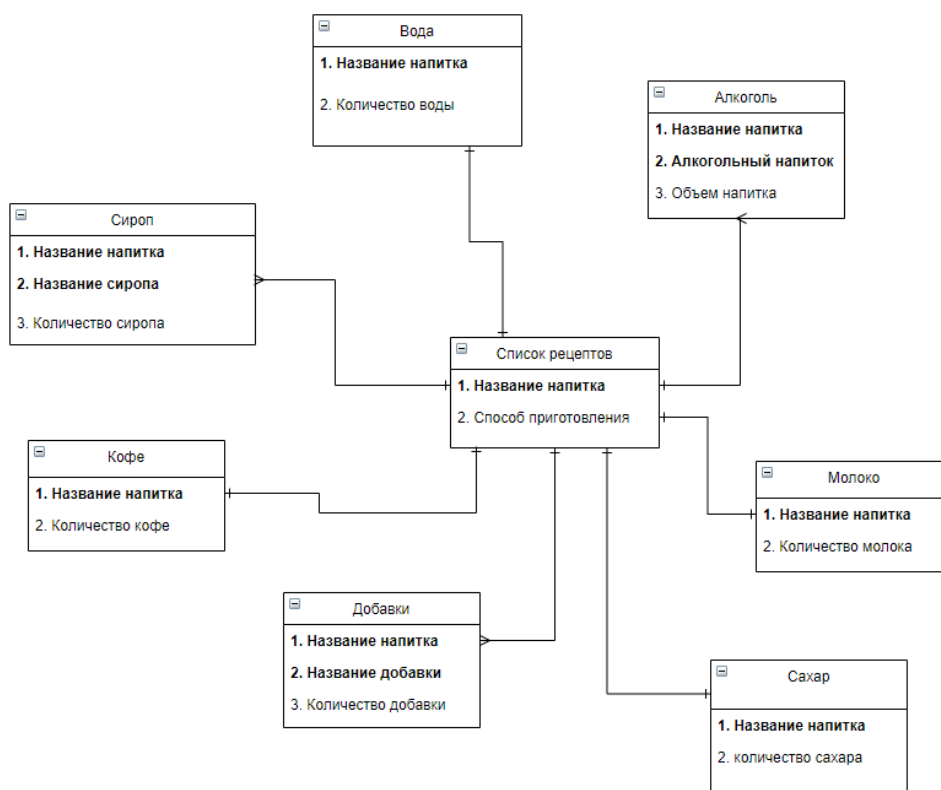


Рис. 1. Концептуальная структура базы данных.

Создание базы данных

Разработка приложения ведется на платформе Unity. Одним из самых популярных способов работы с базами данных на данной платформе является использование SQLite.

SQLite - это библиотека на языке C, которая реализует быстрый, высоконадежный, полнофункциональный механизм базы данных SQL. SQLite встроен во все мобильные телефоны и входит в состав множества приложений, которые люди используют каждый день.

В моем приложении отсутствуют сложные данные, необходимо хранить только числа и текст, а выбранную модель базы данных можно легко реализовать с помощью данной библиотеки. Кроме того, существует большое количество бесплатных утилит и плагинов для создания и редактирования нашей базы, что также упрощает процесс разработки.

Заключение

В результате сравнения двух полученных моделей базы данных, можно сделать вывод, что вторая больше подходит для реализации необходимых функций и возможностей моего приложения. Также она практически не имеет недостатков и соответствует нормальным формам реляционной модели данных, тем самым обеспечивая отсутствие излишней информации и возможных ошибок в процессе работы

Список использованных источников

1. Нормализация базы данных и ее формы. [Электронный ресурс] – URL: <https://office-menu.ru/uroki-sql/51-normalizatsiya-bazy-dannykh> (дата обращения: 08.01.2021).
2. Нормализация отношений. Шесть нормальных форм. [Электронный ресурс] – URL: <https://habr.com/ru/post/254773/> (дата обращения: 07.01.2021).
3. Реляционная модель данных: теоретические основы. [Электронный ресурс]. – URL: https://function-x.ru/sql_relation_data_model.html (дата обращения 05.01.2021).