

ОТЧЁТ

Лабораторная работа №1-1:
«Построение модели данных»

Группа
Студент
Преподаватель

Б21-525
Р.Т. Мясников
М.А. Куприяшин

Оглавление

1.	Предметная область	3
2.	Система таблиц	5
3.	Реализация схемы данных средствами языка SQL в системе управле- ния базами данных SQLite3	8
4.	Заключение	10
5.	Приложение	11

1. Предметная область



В качестве предметной области было выбрано приложение для изучения и запоминания японских иероглифов.

Введение в японскую иероглифику

В японском языке, кроме обычных букв, используются иероглифы. Каждый иероглиф состоит из составных частей - радикалов, которые могут являться как отдельными иероглифами, так и несамостоятельными единицами.

Например, на следующей картинке представлено разложение иероглифа 「有」, который означает "обладать". Он состоит из двух иероглифов 「一」 (один) и 「月」 (луна) и из несамостоятельного радикала - 「丿」:

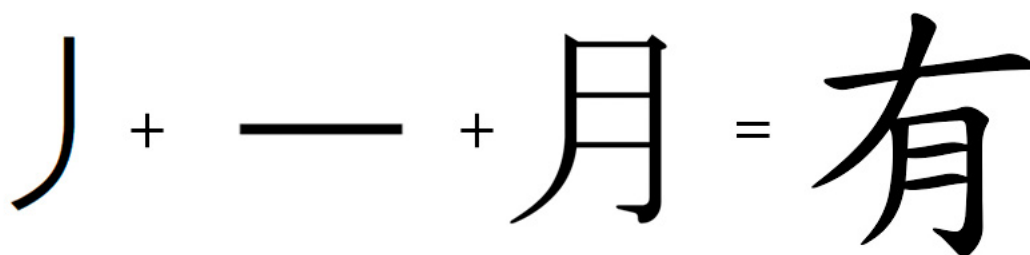


Рис. 1. Пример образования иероглифа

Также важной характеристикой иероглифа является количество штрихов, из которых он составлен, а последовательность иероглифов образуют слова, которые могут как наследовать значения иероглифов, так и полностью не соответствовать им.

Анализ предметной области

Приложение должно предоставлять возможности для эффективного изучения и запоминания иероглифов. Для этого приложение должно поддерживать:

- поиск иероглифов по чтению;
- вывод значения выбранного иероглифа;
- разбиение иероглифов на составные части;
- вывод примеров слов, которые используют выбранный иероглиф;
- поддержка пользователей;
- поддержка пользовательских списков.

2. Система таблиц

Для реализации возможностей приложения была разработана система таблиц для хранения необходимой информации в рамках реляционной базы данных:

- **kanjis** - таблица, перечисляющая иероглифы. В поля таблицы входят: символ иероглифа (**kanji**); значение (**meaning**); радикалы, из которых состоит иероглиф (**radicals_id**); ключ иероглифа (**key**); число штрихов, из которых состоит иероглиф (**strokes**);
- **radicals** - таблица, перечисляющая радикалы. В поля таблицы входят: символ радикала (**radical**); чтение (**reading_id**); значение (**meaning**);
- **words** - таблица, перечисляющая японские слова. В поля таблицы входят: слово (**word**); значение (**meaning**);
- **readings** - таблица, перечисляющая чтения иероглифов и радикалов. В поля таблицы входят: **id**; чтение (**reading**);
- **lists** - таблица, перечисляющая пользовательские списки иероглифов. В поля таблицы входят: **id**; неunikальное название таблицы (**table_name**); владелец таблицы (**username**); описание таблицы (**description**);
- **users** - таблица, перечисляющая пользователей. В поля таблицы входят: **username**.

Отношения сущностей

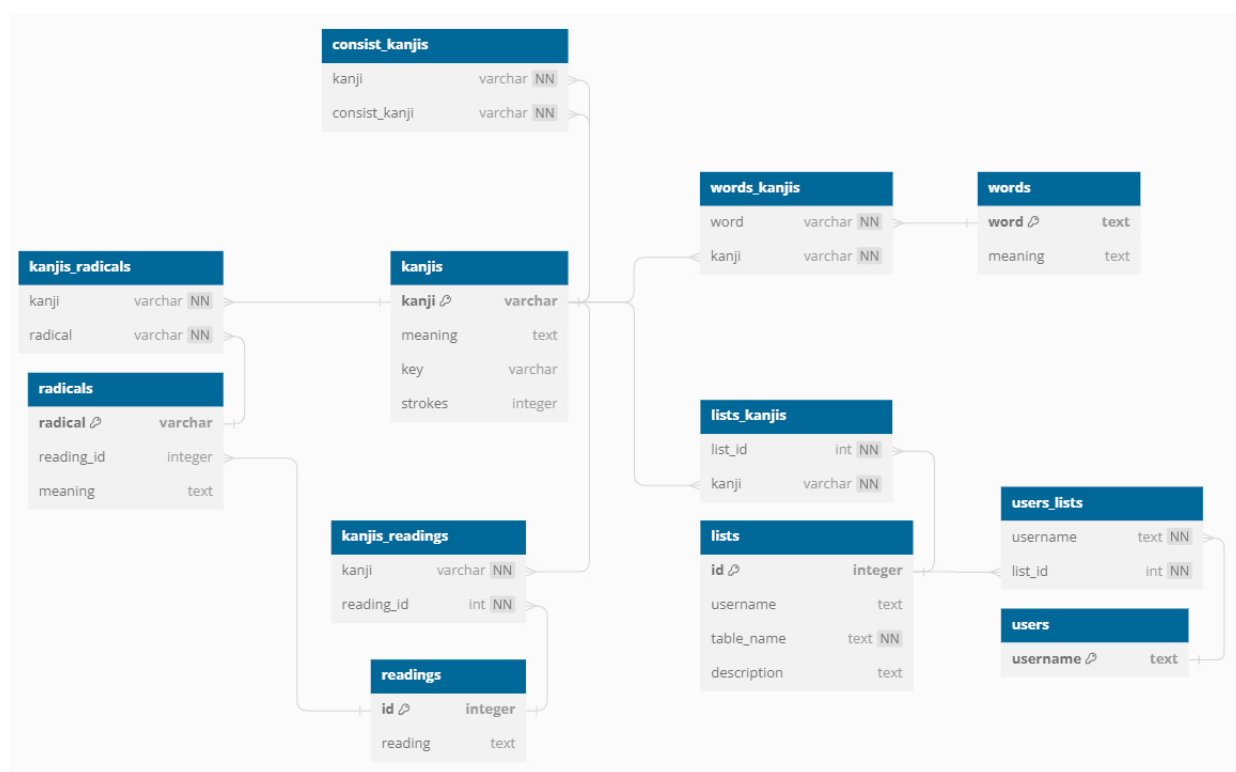


Рис. 2. Система таблиц

Нормальная форма

Чтобы доказать, что ваша база данных находится в третьей нормальной форме (3NF), мы должны убедиться в следующих требованиях:

1. Все неключевые атрибуты каждой таблицы функционально зависят от первичного ключа таблицы.
2. Нет транзитивных зависимостей.

Рассмотрим каждую таблицу по отдельности:

1. Таблица "kanjis":

- "kanji" - первичный ключ;
- "meaning", "key", "strokes" - зависят только от "kanji";
- Все атрибуты зависят от первичного ключа "kanji". Нет транзитивных зависимостей.

2. Таблица "radicals":

- "radical" - первичный ключ;
- "reading_id", meaning - зависят только от "radicals";
- Все атрибуты зависят от первичного ключа "radicals". Нет транзитивных зависимостей.

3. Таблица "words":

- "word" - первичный ключ;
- "meaning" - зависят только от "word".
- Это также соответствует требованиям 3NF.

4. Таблица "readings":

- "id" - первичный ключ;
- "reading" - зависят только от "id".
- Это также соответствует требованиям 3NF.

5. Таблица "users":

- "username" - первичный ключ;
- Это также соответствует требованиям 3NF.

6. Таблицы связей:

- "consist_kanjis", "words_kanjis", "kanjis_radicals", "kanjis_readings", "lists_kanjis", "users_lists" - они все содержат только связи между таблицами, и их первичные ключи состоят из внешних ключей, поэтому они автоматически удовлетворяют требованиям 3NF.

Таким образом, ваша база данных соответствует третьей нормальной форме.

3. Реализация схемы данных средствами языка SQL в системе управления базами данных SQLite3

```
CREATE TABLE kanjis (  
    kanji TEXT PRIMARY KEY,  
    meaning TEXT,  
    key TEXT,  
    strokes INTEGER  
);  
  
CREATE TABLE consist_kanjis (  
    kanji TEXT NOT NULL,  
    consist_kanji TEXT NOT NULL,  
    FOREIGN KEY (kanji) REFERENCES kanjis (kanji),  
    FOREIGN KEY (consist_kanji) REFERENCES kanjis (kanji)  
);  
  
CREATE TABLE kanjis_radicals (  
    kanji TEXT NOT NULL,  
    radical TEXT NOT NULL,  
    FOREIGN KEY (kanji) REFERENCES kanjis (kanji),  
    FOREIGN KEY (radical) REFERENCES radicals (radical)  
);  
  
CREATE TABLE kanjis_readings (  
    kanji TEXT NOT NULL,  
    reading_id INTEGER NOT NULL,  
    FOREIGN KEY (kanji) REFERENCES kanjis (kanji),  
    FOREIGN KEY (reading_id) REFERENCES readings (id)  
);  
  
CREATE TABLE radicals (  
    radical TEXT PRIMARY KEY,  
    reading_id INTEGER,  
    meaning TEXT,  
    FOREIGN KEY (reading_id) REFERENCES readings (id)  
);  
  
CREATE TABLE words (  
    word TEXT PRIMARY KEY,  
    meaning TEXT  
);  
  
CREATE TABLE words_kanjis (  
    word TEXT NOT NULL,  
    kanji TEXT NOT NULL,  
    FOREIGN KEY (word) REFERENCES words (word),  
    FOREIGN KEY (kanji) REFERENCES kanjis (kanji)  
);  
  
CREATE TABLE readings (  
    id INTEGER PRIMARY KEY,  
    reading TEXT UNIQUE
```



```
);

CREATE TABLE users (
    username TEXT PRIMARY KEY
);

CREATE TABLE users_lists (
    username TEXT NOT NULL,
    list_id INTEGER NOT NULL,
    FOREIGN KEY (username) REFERENCES users (username),
    FOREIGN KEY (list_id) REFERENCES lists (id)
);

CREATE TABLE lists (
    id INTEGER PRIMARY KEY,
    username TEXT,
    table_name TEXT NOT NULL,
    description TEXT
);

CREATE TABLE lists_kanjis (
    list_id INTEGER NOT NULL,
    kanji TEXT NOT NULL,
    FOREIGN KEY (list_id) REFERENCES lists (id),
    FOREIGN KEY (kanji) REFERENCES kanjis (kanji)
);
```

Листинг 1. SQLite3 Implementation

4. Заключение

В результате работы были сформулированы требования к базе данных для приложения для изучения и запоминания японских иероглифов. Система таблиц была спроектирована с учетом требований нормализации данных: таблицы удовлетворяют третьей нормальной форме; и с учётом возможностей, которые должно предоставлять приложение. На основе систем таблиц и их зависимости была реализована база данных в системе управлениями баз данных SQLite3.

5. Приложение

Репозиторий: [GitHub](#)