

Архитектура базы данных для Канбан-приложения

1. Назначение базы данных

База данных предназначена для хранения и управления данными многопользовательского канбан-приложения. Основные цели:

- Хранение информации о пользователях, досках, колонках и карточках
- Обеспечение целостности данных при совместной работе
- Реализация системы ролевого доступа
- Поддержка иерархии «доска → колонка → карточка»
- Обеспечение основы для расширения функциональности (теги, комментарии, файлы)

2. Основные сущности и таблицы

2.1. Таблица users (Пользователи)

Назначение: Хранение учётных данных и профилей пользователей.

Поле	Тип	Ограничения	Описание
user_id	INT/BIGINT	PRIMARY KEY, AUTO_INCREMENT	Уникальный идентификатор
email	VARCHAR(255)	UNIQUE, NOT NULL	Электронная почта (логин)
password_hash	VARCHAR(255)	NOT NULL	Хэш пароля
full_name	VARCHAR(100)	NOT NULL	Отображаемое имя
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Дата регистрации
is_active	BOOLEAN	DEFAULT TRUE	Флаг активности

2.2. Таблица boards (Доски)

Назначение: Хранение рабочих пространств (канбан-досок).

Поле	Тип	Ограничения	Описание
board_id	INT/BIGINT	PRIMARY KEY, AUTO_INCREMENT	Уникальный идентификатор доски
name	VARCHAR(100)	NOT NULL	Название доски
format	ENUM('status', 'custom')	DEFAULT 'status'	Формат доски
owner_id	INT/BIGINT NOT NULL	FOREIGN KEY (users.user_id)	Создатель доски
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Дата создания

2.3. Таблица board_members (Участники досок)

Назначение: Связь пользователей с досками и определение ролей (M:M).

Поле	Тип	Ограничения	Описание
board_member_id	INT/BIGINT	PRIMARY KEY, AUTO_INCREMENT	Суррогатный ключ
board_id	INT/BIGINT	FOREIGN KEY (boards.board_id), NOT NULL	Ссылка на доску
user_id	INT/BIGINT	FOREIGN KEY (users.user_id), NOT NULL	Ссылка на пользователя
role	ENUM('creator', 'admin', 'member', 'viewer')	DEFAULT 'member'	Роль на доске
UNIQUE(board_id , user_id)		Уникальная связь	

2.4. Таблица columns (Колонки)

Назначение: Хранение этапов рабочего процесса на доске.

Поле	Тип	Ограничения	Описание
column_id	INT/BIGINT	PRIMARY KEY, AUTO_INCREMENT	Уникальный идентификатор колонки

Поле	Тип	Ограничения	Описание
board_id	INT/BIGINT	FOREIGN KEY (boards.board_id), NOT NULL	Родительская доска
name	VARCHAR(50)	NOT NULL	Название колонки
position	INT	NOT NULL, DEFAULT 0	Порядок отображения

2.5. Таблица cards (Карточки)

Назначение: Хранение задач в виде карточек.

Поле	Тип	Ограничения	Описание
card_id	INT/BIGINT	PRIMARY KEY, AUTO_INCREMENT	Уникальный идентификатор карточки
column_id	INT/BIGINT	FOREIGN KEY (columns.column_id), NOT NULL	Текущая колонка
title	VARCHAR(255)	NOT NULL	Краткое название задачи
description	TEXT	NULLABLE	Подробное описание
created_by	INT/BIGINT	FOREIGN KEY (users.user_id), NOT NULL	Автор карточки
position	INT	NOT NULL, DEFAULT 0	Позиция в колонке
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Дата создания

2.6. Таблица card_assignees (Ответственные за карточки)

Назначение: Связь карточек с ответственными пользователями (М:М).

Поле	Тип	Ограничения	Описание
card_assignee_id	INT/BIGINT	PRIMARY KEY, AUTO_INCREMENT	Суррогатный ключ
card_id	INT/BIGINT	FOREIGN KEY (cards.card_id), NOT NULL	Ссылка на карточку
user_id	INT/BIGINT	FOREIGN KEY (users.user_id), NOT	Ссылка на пользователя

Поле	Тип	Ограничения	Описание
		NULL	
UNIQUE(card_id , user_id)		Уникальная связь	

2.7. Таблица tags (Теги)

Назначение: Хранение цветных тегов для классификации.

Поле	Тип	Ограничения	Описание
tag_id	INT/BIGINT	PRIMARY KEY, AUTO_INCREMENT	Уникальный идентификатор тега
name	VARCHAR(50)	NOT NULL	Текст тега
color	VARCHAR(7)	DEFAULT '#1976D2'	Цвет в HEX-формате
board_id	INT/BIGINT	FOREIGN KEY (boards.board_id), NOT NULL	Привязка к доске

2.8. Таблица card_tags (Теги карточек)

Назначение: Связь карточек с тегами (M:M).

Поле	Тип	Ограничения	Описание
card_tag_id	INT/BIGINT	PRIMARY KEY, AUTO_INCREMENT	Суррогатный ключ
card_id	INT/BIGINT	FOREIGN KEY (cards.card_id), NOT NULL	Ссылка на карточку
tag_id	INT/BIGINT	FOREIGN KEY (tags.tag_id), NOT NULL	Ссылка на тег
UNIQUE(card_id , tag_id)		Уникальная связь	

3. Логические связи между сущностями

Связь	Тип связи	Реализация	Описание
Пользователь ↔ Доска	Многие-ко-многим (M:M)	Через таблицу board_members	Пользователи могут участвовать в нескольких досках с разными ролями
Доска → Колонка	Один-ко-многим (1:M)	Поле board_id в таблице columns	Каждая доска содержит несколько колонок

Связь	Тип связи	Реализация	Описание
Колонка → Карточка	Один-ко-многим (1:M)	Поле <code>column_id</code> в таблице <code>cards</code>	В каждой колонке находится несколько карточек
Карточка ↔ Пользователь	Многие-ко-многим (M:M)	Через таблицу <code>card_assignees</code>	На одну карточку могут быть назначены несколько ответственных
Карточка ↔ Тег	Многие-ко-многим (M:M)	Через таблицу <code>card_tags</code>	Карточка может иметь несколько тегов
Тег → Доска	Многие-к-одному (M:1)	Поле <code>board_id</code> в таблице <code>tags</code>	Тег привязан к конкретной доске

4. Бизнес-правила и ограничения целостности

Каскадные операции

- **Удаление доски** → автоматическое удаление всех связанных колонок, карточек, тегов и записей об участниках
- **Удаление колонки** → автоматическое удаление всех карточек в этой колонке
- **Удаление пользователя** → удаление из всех досок и снятие с ответственных ролей (без удаления досок)

Обязательные связи и ограничения

1. Обязательность родительских сущностей:

- Карточка не может существовать без колонки
- Колонка не может существовать без доски
- Тег должен быть привязан к конкретной доске

2. Система доступа:

- Запись в `board_members` обязательна для доступа к доске
- Без роли доступ к доске запрещён

3. Уровни прав:

- `viewer` (Наблюдатель): только чтение
- `member` (Участник): создание/редактирование карточек
- `admin` (Администратор): управление колонками и участниками
- `creator` (Создатель): полные права (включая удаление доски)

Порядок создания данных

1. `users` → 2. `boards` → 3. `board_members` → 4. `columns` → 5. `cards` → 6. `tags`, `card_assignees`, `card_tags`

5. Обоснование архитектурного решения

Ключевые преимущества

Аспект	Преимущество	Эффект
Бизнес-логика	Прямое соответствие иерархии канбан-методологии	Упрощение разработки и поддержки
Масштабируемость	Легкое добавление новых сущностей	Возможность расширения (комментарии, файлы, история)
Целостность данных	Внешние ключи и каскадные операции	Предотвращение "мусорных" данных
Гибкость управления	Ролевая модель на уровне досок	Тонкая настройка прав доступа
Производительность	Нормализованная структура	Минимизация избыточности данных
Безопасность	Явное управление доступом через <code>board_members</code>	Защита от несанкционированного доступа

Консистентность и надежность

- Все связи защищены внешними ключами
- Каскадные операции обеспечивают согласованность данных
- UNIQUE-ограничения предотвращают дублирование связей
- Индексы на внешние ключи и часто используемые поля оптимизируют запросы

Возможности для расширения

- Добавление таблиц для комментариев (`comments`), файлов (`attachments`), истории изменений (`activity_log`)
- Поддержка подзадач через таблицу `subtasks`
- Интеграция с внешними системами через таблицу `webhooks` или `integrations`