

## Модуль "Пользователи и Авторизация"

### Users

- **Назначение:** Хранение информации о пользователях системы.
- Поля:
  - user\_id: INT, PRIMARY KEY, AUTO\_INCREMENT – Уникальный идентификатор пользователя.
  - name: VARCHAR(255), NOT NULL – Имя пользователя.
  - email: VARCHAR(255), NOT NULL, UNIQUE – Email, используется для входа.
  - password\_hash: VARCHAR(255), NOT NULL – Хеш пароля.
  - role: ENUM('user', 'admin'), NOT NULL, DEFAULT 'user' – Роль пользователя.
  - status: ENUM('inactive', 'active'), NOT NULL, DEFAULT 'inactive' – Статус аккаунта.
  - created\_at: TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP – Дата и время создания.
  - updated\_at: TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP – Дата и время последнего обновления.

### AccountActivationTokens

- **Назначение:** Хранение токенов для активации новых учетных записей.
- Поля:
  - token: VARCHAR(255), PRIMARY KEY – Уникальный токен.
  - user\_id: INT, NOT NULL, FOREIGN KEY -> Users(user\_id) – Ссылка на пользователя.
  - expires\_at: TIMESTAMP, NOT NULL – Срок действия токена.

## Модуль "Мониторинг пасеки"

### Apiaries

- **Назначение:** Хранение информации о пасеках пользователя.
- Поля:
  - apiary\_id: INT, PRIMARY KEY, AUTO\_INCREMENT – Идентификатор пасеки.
  - user\_id: INT, NOT NULL, FOREIGN KEY -> Users(user\_id) – Владелец пасеки.
  - name: VARCHAR(255), NOT NULL – Название пасеки.
  - location: TEXT – Описание местоположения или координаты.
  - created\_at: TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP – Дата создания.

### Hives

- **Назначение:** Хранение информации об ульях на пасеке.
- Поля:
  - hive\_id: INT, PRIMARY KEY, AUTO\_INCREMENT – Идентификатор улья.
  - apiary\_id: INT, NOT NULL, FOREIGN KEY -> Apiaries(apiary\_id) – Пасека, к которой относится улей.
  - name: VARCHAR(255), NOT NULL – Название или номер улья.
  - created\_at: TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP – Дата добавления.

### BeeColonies

- **Назначение:** Хранение данных о пчелосемьях в ульях.
- Поля:
  - colony\_id: INT, PRIMARY KEY, AUTO\_INCREMENT – Идентификатор пчелосемьи.
  - hive\_id: INT, NOT NULL, FOREIGN KEY -> Hives(hive\_id) – Улей, в котором находится семья.
  - breed: VARCHAR(255) – Порода пчел.
  - queen\_age: INT – Возраст матки (в месяцах).
  - installation\_date: DATE – Дата заселения семьи в улей.
  - status: ENUM('active', 'archived'), NOT NULL – Статус (текущая или архивная).
  - notes: TEXT – Заметки пчеловода.

### Sensors

- **Назначение:** Реестр датчиков, установленных на ульях.
- Поля:
  - sensor\_id: INT, PRIMARY KEY, AUTO\_INCREMENT – Идентификатор датчика.
  - hive\_id: INT, NOT NULL, FOREIGN KEY -> Hives(hive\_id) – Улей, на котором установлен датчик.
  - device\_identifier: VARCHAR(255), NOT NULL, UNIQUE – Уникальный идентификатор физического устройства.
  - sensor\_type: ENUM('weight', 'temperature', 'humidity', 'sound'), NOT NULL – Тип датчика.
  - status: ENUM('active', 'inactive', 'error'), NOT NULL – Рабочий статус датчика.
  - installed\_at: TIMESTAMP – Дата установки.

## SensorReadings

- **Назначение:** Хранение временных рядов данных, поступающих с датчиков. Это самая большая таблица.
- Поля:
  - reading\_id: BIGINT, PRIMARY KEY, AUTO\_INCREMENT – Идентификатор показания.
  - sensor\_id: INT, NOT NULL, FOREIGN KEY -> Sensors(sensor\_id).
  - value: DECIMAL(10, 2), NOT NULL – Значение показания.
  - timestamp: TIMESTAMP WITH TIME ZONE, NOT NULL – Точное время снятия показания.
- **Индексы:** Композитный индекс по (sensor\_id, timestamp) для быстрого извлечения данных для графиков.

## Alerts

- **Назначение:** Хранение сгенерированных системой предупреждений.
- Поля:
  - alert\_id: INT, PRIMARY KEY, AUTO\_INCREMENT – Идентификатор оповещения.
  - user\_id: INT, NOT NULL, FOREIGN KEY -> Users(user\_id).
  - hive\_id: INT, NOT NULL, FOREIGN KEY -> Hives(hive\_id).
  - alert\_type: VARCHAR(255), NOT NULL – Тип предупреждения (например, "possible\_swarm").
  - description: TEXT – Описание.
  - status: ENUM('new', 'read', 'resolved'), NOT NULL – Статус.
  - timestamp: TIMESTAMP, NOT NULL – Время генерации.

## Модуль "База Знаний"

### ArticleCategories

- **Назначение:** Хранение категорий для статей.
- Поля:
  - category\_id: INT, PRIMARY KEY, AUTO\_INCREMENT – Идентификатор категории.
  - parent\_category\_id: INT, FOREIGN KEY -> ArticleCategories(category\_id) – Для создания иерархии (может быть NULL).
  - name: VARCHAR(255), NOT NULL, UNIQUE – Название категории.
  - description: TEXT – Описание.

### Articles

- **Назначение:** Хранение статей базы знаний.
- Поля:
  - article\_id: INT, PRIMARY KEY, AUTO\_INCREMENT – Идентификатор статьи.
  - category\_id: INT, NOT NULL, FOREIGN KEY -> ArticleCategories(category\_id).
  - author\_id: INT, NOT NULL, FOREIGN KEY -> Users(user\_id).
  - title: VARCHAR(255), NOT NULL – Заголовок.
  - content: TEXT, NOT NULL – Содержимое статьи.
  - status: ENUM('draft', 'published', 'archived'), NOT NULL – Статус публикации.
  - created\_at: TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP.
  - updated\_at: TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP.
- **Индексы:** Полнотекстовый индекс по полям title и content для реализации поиска.

### Comments

- **Назначение:** Хранение комментариев к статьям.
- Поля:
  - comment\_id: INT, PRIMARY KEY, AUTO\_INCREMENT – Идентификатор комментария.
  - article\_id: INT, NOT NULL, FOREIGN KEY -> Articles(article\_id).
  - user\_id: INT, NOT NULL, FOREIGN KEY -> Users(user\_id).
  - parent\_comment\_id: INT, FOREIGN KEY -> Comments(comment\_id) – Для вложенных комментариев (может быть NULL).
  - content: TEXT, NOT NULL – Текст комментария.
  - created\_at: TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP.

Для корректной работы базы данных с учетом внешних ключей (FOREIGN KEY) таблицы следует заполнять в следующем порядке:

1. **Users:** Сначала создаются пользователи, особенно администраторы, которые будут добавлять контент.
2. **ArticleCategories:** Создаются категории для статей.

3. **Articles**: Статьи можно добавлять после создания пользователей (авторов) и категорий.
4. **Comments**: Комментарии добавляются к существующим статьям существующими пользователями.
5. **Apiaries**: Пользователи могут создавать свои пасеки.
6. **Hives**: Ульи добавляются в существующие пасеки.
7. **BeeColonies**: Информация о пчелосемьях добавляется для конкретных ульев.
8. **Sensors**: Датчики привязываются к ульям.
9. **SensorReadings**: Данные с датчиков начинают поступать и сохраняться.
10. **Alerts**: Оповещения генерируются для пользователей по данным из их ульев.
11. **AccountActivationTokens**: Записи создаются в процессе регистрации нового пользователя.