

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.