**1. Сущности и структура БД (PostgreSQL)**

**1.1. Таблицы и их поля**

**1. Пользователи (**users**)**

| **Поле** | **Тип данных** | **Описание** | **Ограничения** |
| --- | --- | --- | --- |
| id | SERIAL/UUID | Уникальный идентификатор | PRIMARY KEY |
| username | VARCHAR(50) | Логин | UNIQUE, NOT NULL |
| email | VARCHAR(100) | Email | UNIQUE, NOT NULL |
| password | VARCHAR(255) | Хешированный пароль | NOT NULL |
| is\_admin | BOOLEAN | Права администратора | DEFAULT FALSE |
| rating | INTEGER | Рейтинг пользователя | DEFAULT 0 |
| created\_at | TIMESTAMP | Дата регистрации | DEFAULT NOW() |

**2. Проблемы (**issues**)**

| **Поле** | **Тип данных** | **Описание** | **Ограничения** |
| --- | --- | --- | --- |
| id | SERIAL/UUID | Уникальный ID проблемы | PRIMARY KEY |
| user\_id | INTEGER/UUID | Автор заявки | FOREIGN KEY (users.id) |
| type | VARCHAR(50) | Тип проблемы (яма, мусор и т.д.) | NOT NULL |
| short\_desc | VARCHAR(200) | Краткое описание | NOT NULL |
| full\_desc | TEXT | Подробное описание | NOT NULL |
| status | VARCHAR(20) | Статус (новая/в обработке/решена) | DEFAULT 'new' |
| address | VARCHAR(255) | Адрес проблемы | NOT NULL |
| latitude | FLOAT | Широта (для карты) |  |
| longitude | FLOAT | Долгота (для карты) |  |
| created\_at | TIMESTAMP | Дата создания | DEFAULT NOW() |
| updated\_at | TIMESTAMP | Дата обновления статуса | DEFAULT NOW() |

**3. Фотографии (**photos**)**

| **Поле** | **Тип данных** | **Описание** | **Ограничения** |
| --- | --- | --- | --- |
| id | SERIAL/UUID | Уникальный ID фото | PRIMARY KEY |
| issue\_id | INTEGER/UUID | ID связанной проблемы | FOREIGN KEY (issues.id) |
| file\_path | VARCHAR(255) | Путь к файлу (S3 или локально) | NOT NULL |
| uploaded\_at | TIMESTAMP | Дата загрузки | DEFAULT NOW() |

**4. Проблемы (**issues\_field**)**

| **Поле** | **Тип данных** | **Описание** | **Ограничения** |
| --- | --- | --- | --- |
| id | SERIAL/UUID | Уникальный ID проблемы | PRIMARY KEY |
| type | VARCHAR(50) | Тип проблемы (яма, мусор и т.д.) | NOT NULL |

**3. Связи между таблицами**

* **1 пользователь → N проблем** (users.id → issues.user\_id).
* **1 проблема → N фото** (issues.id → photos.issue\_id).

**4. Индексы для ускорения запросов???**

**5. Требования к безопасности**

1. **Шифрование**:
   * Пароли хранятся в виде хешей (bcrypt/scrypt).
   * Персональные данные (email) шифруются при необходимости (PGP).
2. **Защита от SQL-инъекций**:
   * Использование ORM (SQLAlchemy) или подготовленных запросов.

**6. Производительность и масштабируемость**

* **Лимиты**:
  + Максимальный размер фото — 5 МБ (проверка на уровне NGINX и API).
  + Ограничение запросов (rate limiting) для API

**8. SQL-запросы**

Получение всех проблем пользователя

Фильтрация проблем по статусу

Обновление статуса проблемы

Запросы/представления для аналитики:

1. Количество проблем по категориям(пример):

* + Дорожные ямы: 125
  + Неубранный мусор: 87
  + Освещение: 45
  + Поврежденные здания: 32
  + Нарушения парковки: 50

2. По статусу

Новые: 62

В обработке: 98

Решено: 179

3. По районам

4. По дате времени

**9.Триггеры**

1. Триггер для обновления времени статуса