1. ***Описание структуры разработанной БД***.

Краткое описание таблиц базы данных и связей.

1. **users** – пользователи платформы.
   * id (PK) – уникальный идентификатор пользователя.
   * name – имя пользователя.
   * email – электронная почта.
   * password – хешированный пароль.
   * preferences – предпочтения пользователя.
2. **chefs** – кулинарные мастера.
   * id (PK) – уникальный идентификатор мастера.
   * name – имя мастера.
   * bio – описание.
   * specialization – специализация.
3. **recipes** – рецепты.
   * id (PK) – уникальный идентификатор рецепта.
   * title – название.
   * description – описание.
   * chef\_id (FK) – автор рецепта.
4. **ingredients** – ингредиенты.
   * id (PK) – уникальный идентификатор ингредиента.
   * name – название.
5. **recipe\_ingredients** – связь рецептов и ингредиентов (многие-ко-многим).
   * recipe\_id (FK) – ссылка на рецепт.
   * ingredient\_id (FK) – ссылка на ингредиент.
   * quantity – количество.
6. **masterclasses** – расписание мастер-классов.
   * id (PK) – уникальный идентификатор мастер-класса.
   * title – название.
   * date\_time – дата и время.
   * chef\_id (FK) – ведущий мастер-класса.
7. **registrations** – записи пользователей на мастер-классы.
   * user\_id (FK) – пользователь.
   * masterclass\_id (FK) – мастер-класс.
8. Включите ER-диаграмму и SQL-дамп для развертывания базы данных.

CREATE TABLE users (

id SERIAL PRIMARY KEY,

name VARCHAR(100) NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

password VARCHAR(255) NOT NULL,

preferences TEXT

);

CREATE TABLE chefs (

id SERIAL PRIMARY KEY,

name VARCHAR(100) NOT NULL,

bio TEXT,

specialization VARCHAR(100)

);

CREATE TABLE recipes (

id SERIAL PRIMARY KEY,

title VARCHAR(255) NOT NULL,

description TEXT,

chef\_id INT REFERENCES chefs(id) ON DELETE CASCADE

);

CREATE TABLE ingredients (

id SERIAL PRIMARY KEY,

name VARCHAR(100) UNIQUE NOT NULL

);

CREATE TABLE recipe\_ingredients (

recipe\_id INT REFERENCES recipes(id) ON DELETE CASCADE,

ingredient\_id INT REFERENCES ingredients(id) ON DELETE CASCADE,

quantity VARCHAR(50),

PRIMARY KEY (recipe\_id, ingredient\_id)

);

CREATE TABLE masterclasses (

id SERIAL PRIMARY KEY,

title VARCHAR(255) NOT NULL,

date\_time TIMESTAMP NOT NULL,

chef\_id INT REFERENCES chefs(id) ON DELETE SET NULL

);

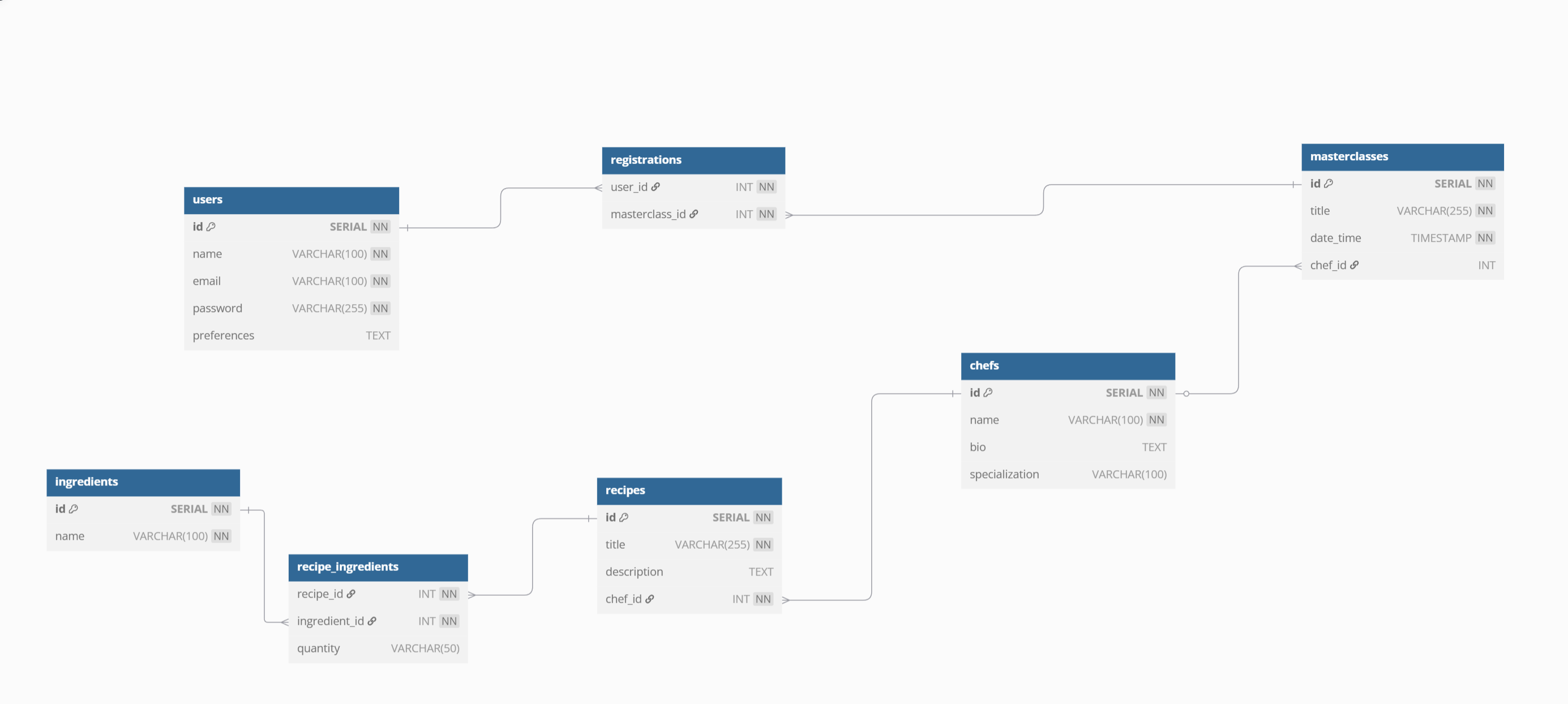
CREATE TABLE registrations (

user\_id INT REFERENCES users(id) ON DELETE CASCADE,

masterclass\_id INT REFERENCES masterclasses(id) ON DELETE CASCADE,

PRIMARY KEY (user\_id, masterclass\_id)

);



1. ***Документирование API***. Список всех маршрутов API с методами, описанием, параметрами и кодами ответа.

| **Маршрут** | **Метод** | **Описание** | **Параметры в body** | **Заголовки** | **Коды ответа** |
| --- | --- | --- | --- | --- | --- |
| /api/users | POST | Регистрация пользователя | { "name": "John", "email": "john@example.com", "password": "secure123" } | Content-Type: application/json | 201: Создано, 400: Ошибка валидации |
| /api/users/login | POST | Авторизация пользователя | { "email": "john@example.com", "password": "secure123" } | Content-Type: application/json | 200: Успех, 401: Неверные данные |
| /api/chefs | GET | Получение списка кулинарных мастеров | - | Content-Type: application/json | 200: Успех |
| /api/recipes | GET | Получение всех рецептов | - | Content-Type: application/json | 200: Успех |
| /api/recipes | POST | Создание нового рецепта | { "title": "Паста", "description": "Итальянская паста", "chef\_id": 1 } | Content-Type: application/json | 201: Создано, 400: Ошибка |
| /api/recipes/{id} | GET | Получение рецепта по ID | - | Content-Type: application/json | 200: Успех, 404: Не найдено |
| /api/recipes/{id} | PUT | Обновление рецепта | { "title": "Обновленная паста" } | Content-Type: application/json | 200: Успех, 400: Ошибка |
| /api/recipes/{id} | DELETE | Удаление рецепта | - | Content-Type: application/json | 204: Удалено, 404: Не найдено |
| /api/masterclasses | GET | Получение списка мастер-классов | - | Content-Type: application/json | 200: Успех |
| /api/masterclasses/{id}/register | POST | Запись на мастер-класс | { "user\_id": 2 } | Content-Type: application/json | 201: Успешно записан, 400: Ошибка |
| /api/shopping-list | POST | Генерация списка покупок по рецепту | { "recipe\_id": 5 } | Content-Type: application/json | 200: Успех, 400: Ошибка |
| /api/recommendations | GET | Получение рекомендаций мастер-классов | - | Content-Type: application/json | 200: Успех |

1. Ссылка на код