

Пусть требуется разработать информационную систему для автоматизации учёта рецептов блюд.

Система должна предусматривать хранение данных о блюдах, ингредиентах, типах блюд, а также о составе каждого блюда и характеристиках его приготовления. Система должна обеспечивать возможность поиска, фильтрации и анализа рецептов по различным параметрам, таким как тип блюда, сезон, калорийность, необходимое оборудование и т. д.

В системе ведётся несколько таблиц, отражающих основные сущности предметной области: «Ингредиенты», «Типы блюд», «Блюда» и «Состав».

**Таблица «Ингредиенты»** предназначена для хранения информации обо всех продуктах и веществах, используемых при приготовлении блюд. Каждый ингредиент характеризуется следующими параметрами:

- уникальный идентификатор (первичный ключ);
- название ингредиента;
- примечание (например, особенности хранения, сезонность, допустимые замены и т. д.).

**Таблица «Типы блюд»** предназначена для классификации всех блюд, находящихся в базе данных.

Для каждого типа блюда указывается его уникальное название, которое является первичным ключом.

Примеры типов блюд: первое, гарнир, десерт, салат, напиток и т. д.

**Таблица «Блюда»** содержит сведения обо всех рецептах, хранящихся в системе.

Каждое блюдо характеризуется следующими параметрами:

- идентификатор — уникальный номер (первичный ключ);
- название блюда (обязательное поле);
- тип блюда (внешний ключ, ссылающийся на таблицу «Типы блюд»);
- общий вес блюда в граммах (обязательное поле);
- количество порций (по умолчанию 1);
- сезон (список возможных значений: лето, зима, все, весна-лето и т. д.);
- необходимое оборудование (например, плита, духовка, микроволновая печь и т. д.);
- калорийность на 100 грамм продукта;
- время приготовления (обязательное поле);
- последовательность приготовления (обязательное текстовое поле, содержащее описание шагов приготовления блюда).

Каждое блюдо может включать в себя множество ингредиентов, а один и тот же ингредиент может входить в состав различных блюд.

Одно и то же блюдо может иметь одинаковое название с другим, однако все блюда различаются по уникальному идентификатору.

**Таблица «Состав»** предназначена для описания состава каждого блюда. Для каждого блюда указывается перечень ингредиентов, их количество и единицы измерения.

Таблица имеет следующие поля:

- блюдо (внешний ключ, ссылающийся на таблицу «Блюда»);
- ингредиент (внешний ключ, ссылающийся на таблицу «Ингредиенты»);
- количество ингредиента;
- единица измерения (например, г, мл, шт и т. д.).

#### **Ограничения, предусмотренные в системе:**

1. Одно блюдо не может содержать дважды один и тот же ингредиент.
2. Поля «название» и «последовательность приготовления» обязательны для заполнения.
3. Количество порций не может быть меньше 1.
4. В системе должны храниться только блюда, для которых указано время приготовления.
5. Если для приготовления блюда не требуется оборудование, поле «необходимое оборудование» может быть пустым.
6. Тип блюда должен соответствовать одному из значений, записанных в таблице «Типы блюд».
7. Вес порции вычисляется автоматически как отношение общего веса блюда к количеству порций.

#### **Система должна обеспечивать выполнение следующих функций:**

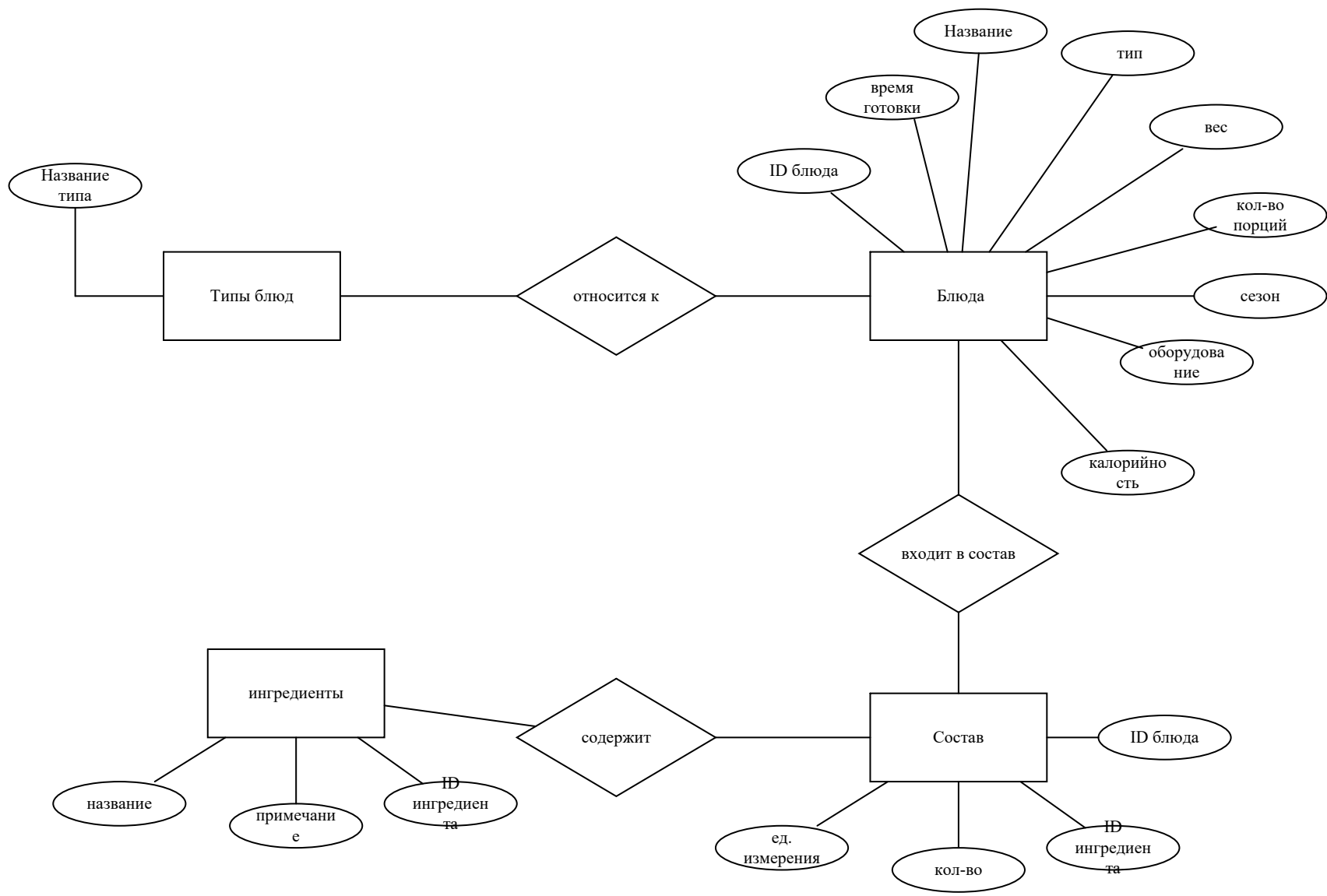
1. Ведение справочников ингредиентов и типов блюд.
2. Регистрацию новых блюд с указанием их характеристик и состава.
3. Изменение и удаление рецептов, а также редактирование состава блюда.
4. Поиск блюд по различным критериям (по типу, по ингредиенту, по сезону, по калорийности, по времени приготовления и т. д.).
5. Выборку данных по различным условиям. В частности, должны быть сформированы следующие запросы:
  - список летних салатов, время приготовления которых меньше получаса;
  - список блюд, в которые входит авокадо;
  - список десертов, не требующих применения оборудования;
  - список блюд, для которых вес одной порции меньше 50 г или больше 250 г;
  - список ингредиентов, которые не входят ни в одно блюдо.

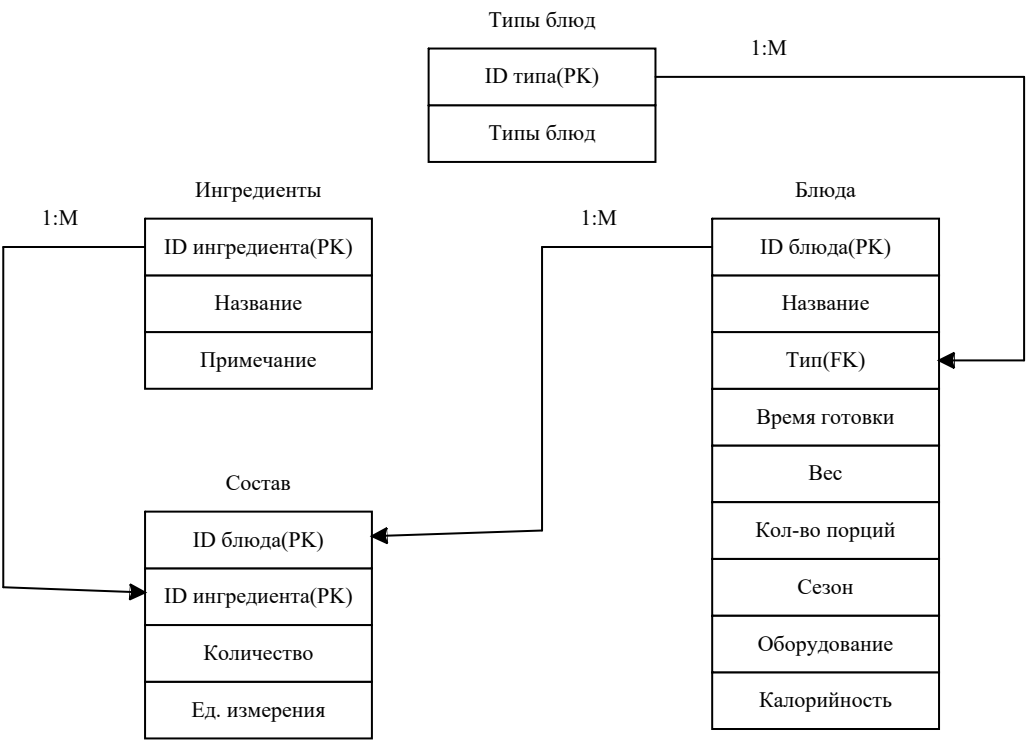
Один из запросов должен быть реализован двумя способами с последующим сравнением скорости выполнения и объяснением, какой вариант будет работать быстрее и почему.

### **Представления, предусмотренные в системе:**

1. **Представление «Калорийность одной порции»** — содержит данные о блюде, весе одной порции и калорийности. Если количество порций не указано, то вес одной порции считается равным 100 граммам.
2. **Представление «Состав блюда»** — содержит название блюда, ингредиент, количество и единицу измерения.
3. **Представление «Мясные блюда, для приготовления которых необходима духовка»** — содержит информацию о тех блюдах, которые включают мясные ингредиенты и требуют использования духовки.

Для каждого созданного представления необходимо проверить возможность выполнения операций обновления (UPDATE), удаления (DELETE) и вставки (INSERT), и объяснить, являются ли данные представления обновляемыми.





```
CREATE DATABASE IF NOT EXISTS food_schema;
```

```
USE food_schema;
```

```
CREATE TABLE Types (  
    ID_type INT AUTO_INCREMENT PRIMARY KEY,  
    Type_name VARCHAR(100) NOT NULL  
);
```

```
CREATE TABLE Ingredients (  
    ID_ingredient INT AUTO_INCREMENT PRIMARY KEY,  
    Name VARCHAR(150) NOT NULL,  
    Note VARCHAR(255)  
);
```

```
CREATE TABLE Dishes (  
    ID_dish INT AUTO_INCREMENT PRIMARY KEY,  
    Name VARCHAR(150) NOT NULL,  
    Type INT NOT NULL,  
    Cooking_time INT,  
    Weight INT,  
    Portions INT,  
    Season VARCHAR(100),  
    Equipment VARCHAR(150),  
    Calories INT,  
    FOREIGN KEY (Type) REFERENCES Types(ID_type)  
);
```

```
CREATE TABLE Recipe (  
    ID_dish INT NOT NULL,  
    ID_ingredient INT NOT NULL,  
    Quantity DECIMAL(10,2),
```

```
Unit VARCHAR(30),  
PRIMARY KEY (ID_dish, ID_ingredient),  
FOREIGN KEY (ID_dish) REFERENCES Dishes(ID_dish),  
FOREIGN KEY (ID_ingredient) REFERENCES Ingredients(ID_ingredient)  
);
```