**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

**по дисциплине «Базы данных»**

Вариант 22

|  |  |  |
| --- | --- | --- |
| Студент гр. 4381 |  | Сергеев А.Д. |
| Преподаватель |  | Фомичева Т.Г. |

Санкт-Петербург

2020

**содержание**

[1. Постановка задачи 3](#_Toc59598466)

[1.1. Задание. 3](#_Toc59598467)

[2. Проектирование БД 5](#_Toc59598468)

[2.1. Описание сущностей 5](#_Toc59598469)

[2.2. ER-модель 5](#_Toc59598470)

[2.3. Реляционная модель 7](#_Toc59598471)

[2.4. Проверка соответствия НФБК 9](#_Toc59598472)

[3. Структуры таблиц и постоянные связи между таблицами 14](#_Toc59598473)

[3.1. Структуры таблиц 14](#_Toc59598474)

[4. Содержимое таблиц 17](#_Toc59598475)

[5. Схема иерархии интерфейса 22](#_Toc59598476)

[6. Описание экранных форм 23](#_Toc59598477)

[6.1. Основа 23](#_Toc59598478)

[6.2. Заказы 24](#_Toc59598479)

[6.3. Заказ 25](#_Toc59598480)

[6.4. Официанты 28](#_Toc59598481)

[6.5. Официант 29](#_Toc59598482)

[6.6. Контракт 30](#_Toc59598483)

[6.7. Финансовые отчеты 31](#_Toc59598484)

[7. Описание запросов 32](#_Toc59598485)

[7.1. Добавить блюдо в заказ 32](#_Toc59598486)

[7.2. Получить продукты для заказа 32](#_Toc59598487)

[7.3. Проверить доступные блюда 33](#_Toc59598488)

[7.4. Закупить продукты на кухню 34](#_Toc59598489)

[7.5. Приготовить заказ 34](#_Toc59598490)

[8. Описание отчетов 35](#_Toc59598491)

[8.1. Меню 35](#_Toc59598492)

[8.2. Блюдо 35](#_Toc59598493)

[8.3. Финансовые отчеты 38](#_Toc59598494)

[8.4. Заказ 39](#_Toc59598495)

[8.5. Чек 40](#_Toc59598496)

[Заключение 41](#_Toc59598501)

# Постановка задачи

## Задание.

Пусть требуется создать программную систему, предназначенную для функционирования ресторана. Такая система должна обеспечивать хранение сведений о ресторане, о наличии на его кухне продуктов, необходимых для приготовления блюд, о поставщиках и продуктах, закупаемых у этих поставщиков. Ресторан осуществляет закупку продуктов у разных поставщиков, предпочитая при этом закупать одни виды продуктов у одних поставщиков, а другие у других. Также необходимо обеспечивать хранение информации о работающих в ресторане, о блюдах, которое готовят в данном заведении, заказах, которые когда-либо делались здесь и о содержимом данных заказов.

Пусть количество столиков в ресторане известно, и имеются официанты, которые обслуживают их. Официанты могут обслуживать любые столики, без какой-либо привязки. При входе в ресторан клиенты могут выбрать один столик, за который бы они могли сесть. При заказе блюд, необходимо указывать имя официанта, который бы их обслуживал. О каждом заказе должна храниться следующая информация: номер столика, за которым сидел/и посетители, блюда, которые они заказывали, имя официанта, обслуживающего их, дата и время заказа. Также необходимо предусмотреть хранение всей информации о поставки продуктов питания, когда-либо осуществляемой в данном ресторане.

Работа с системой предполагает реализацию следующих возможностей:

* Выбор пользователем блюд из меню
* Принятие заказов блюд у посетителей ресторана
* Формирование чека для посетителя
* Осуществление заказа поставки продуктов питания
* Получение информации о любом заказе, сделанном когда-либо в данном заведение
* Принятие / увольнение работников
* Просмотра отчета о доходах / расходах ресторана за указанный период
* Получение информации о количестве продуктов на складе и порций блюд, которые можно приготовить из них

# Проектирование БД

Для проектирования базы данных выбран ER-метод (метод "сущность-связь").

При использовании этого метода необходимо прежде всего создать ER- модель, отражающую связи сущностей заданной предметной области.

Описание сущностей может включать в себя только ключевые атрибуты сущностей – их уникальные свойства, необходимых для идентификации экземпляра сущности.

## Описание сущностей

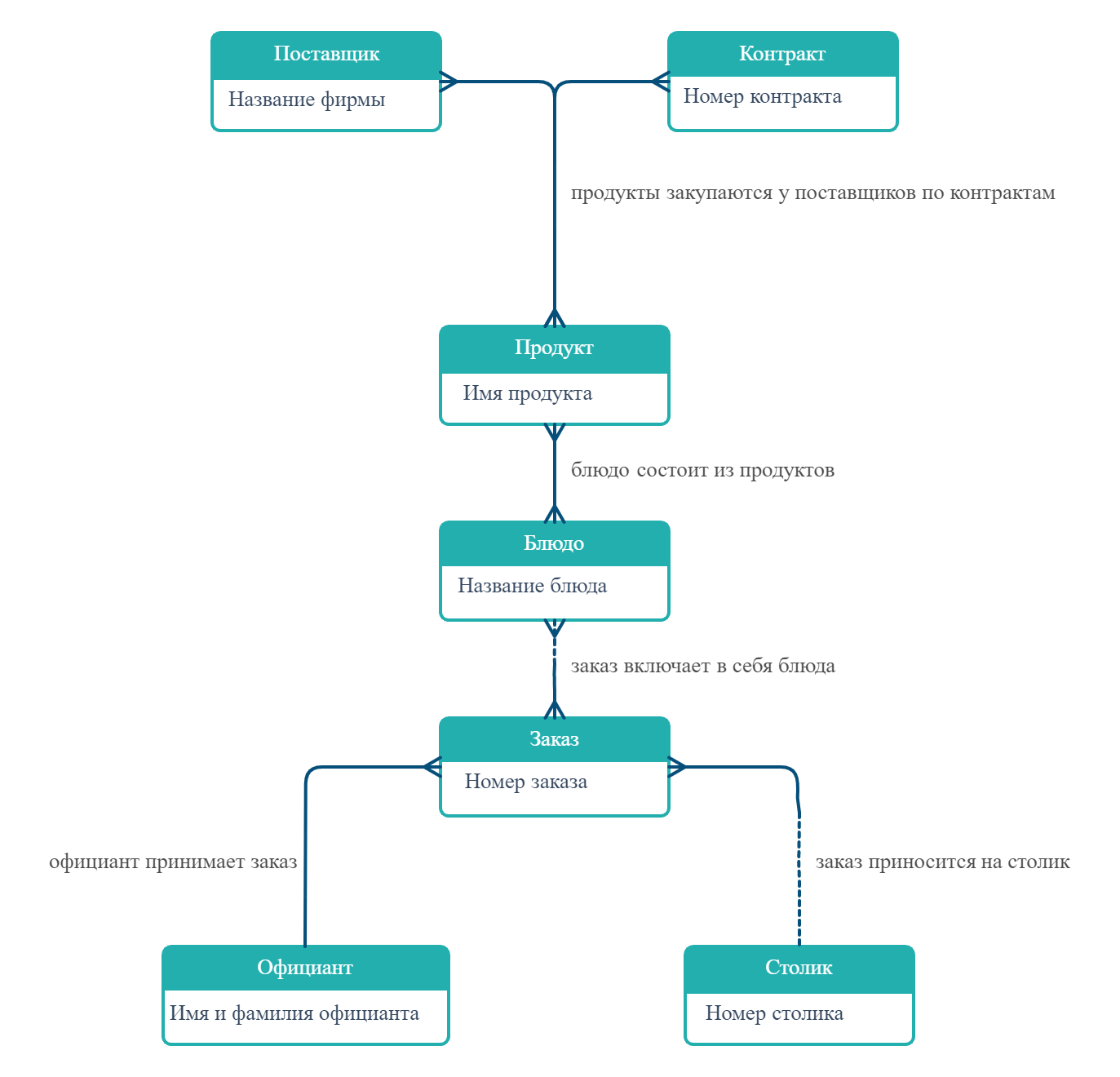
Можно выделить следующие сущности:

* «Официант» с ключом «Имя и фамилия официанта» и атрибутами: «Номер телефона», «Должность», «Работает» (атрибут содержит информацию о том, работает ли официант в настоящий момент или находится на больничном / в отпуске / уволен).
* «Столик» с ключом «Номер столика» и атрибутами: «Название зала», «Количество мест».
* «Заказ» с ключом «Номер заказа» и атрибутом «Дата» (атрибут содержит информацию о точной дате и времени совершения заказа).
* «Блюдо» с ключом «Название блюда» и атрибутами: «Пищевая ценность», «Вес», «Цена».
* «Продукт» с ключом «Имя продукта» и атрибутами: «На кухне», «Единицы измерения».
* «Поставщик» с ключом «Название фирмы».
* «Контракт» с ключом «Номер контракта» и атрибутами: «Сумма», «Дата».

## ER-модель

Между сущностями можно установить следующие связи:

* «Официант - Заказ»: каждый заказ должен быть принят каким-либо официантом, каждый официант принимает заказы. Связь обязательная с обеих сторон, имеет степень 1 со стороны официанта и n со стороны заказа.
* «Столик - Заказ»: каждый заказ приносится на определенный столик, на каждый столик в разное время может быть принесено несколько разных заказов, в то же время какой-нибудь столик может не обслуживаться (например, столики на улице зимой). Связь обязательная со стороны заказа и не обязательная со стороны столика, имеет степень 1 со стороны столика и n со стороны заказа.
* «Заказ - Блюдо»: каждый заказ включает в себя одно или несколько блюд, каждое блюдо может присутствовать в нескольких заказах. Связь обязательная со стороны заказа и не обязательная со стороны блюда, имеет степень n с обеих сторон.
* «Блюдо - Продукт»: каждое блюдо состоит из одного или нескольких продуктов, а каждый продукт входит в состав одного или нескольких блюд. Связь обязательная с обеих сторон, имеет степень n с обеих сторон.
* «Продукт – Контракт - Поставщик»: каждый продукт закупается у одного или нескольких поставщиков по одному или нескольким контрактам, каждый поставщик продает один или несколько продуктов по одному или нескольким контрактам, а каждый контракт включает в себя один или несколько продуктов, купленных у одного или нескольких поставщиков. Связь обязательная с трех сторон, имеет степень n с трех сторон.



Составив ER-модель, можем перейти к созданию реляционной модели базы данных.

## Реляционная модель

Связь **"Заказ" – "Официант"** образует два отношения – по одному для каждой сущности. В отношение "Заказ" добавляются ключевые атрибуты сущности "Официант":

* Заказ (Номер заказа, Имя официанта).
* Официант (Имя и фамилия официанта).

Связь **"Заказ" – "Столик"** образует два отношения – по одному для каждой сущности. В отношение "Заказ" добавляются ключевые атрибуты сущности "Столик":

* Заказ (Номер заказа, Номер столика).
* Столик (Номер столика).

Связь **"Заказ" – "Блюдо"** образует три отношения – два объектных и одно связное. Связное отношение "Заказано" содержит ключевые атрибуты сущностей "Заказ" и "Блюдо":

* Заказано (Номер заказа, Название блюда).
* Заказ (Номер заказа).
* Блюдо (Название блюда).

Связь **"Блюдо" – "Продукт"** образует три отношения – два объектных и одно связное. Связное отношение "Содержит" содержит ключевые атрибуты сущностей "Блюдо" и "Продукт", а также атрибут, определяет количество продукта в блюде:

* Содержит (Название блюда, Имя продукта, Количество продукта).
* Блюдо (Название блюда).
* Продукт (Имя продукта).

Связь **"Поставщик" – "Продукт" – "Контракт"** образует четыре отношения – три объектных и одно связное. Связное отношение "Закуплено" содержит ключевые атрибуты сущностей "Поставщик", "Продукт" и "Контракт", а также атрибут, определяющий количество закупленного продукта:

* Закуплено (Номер контракта, Имя продукта, Название фирмы, Количество).
* Поставщик (Название фирмы).
* Продукт (Имя продукта).
* Контракт (Номер контракта).

Распределим между полученными отношениями оставшиеся атрибуты предметной области:

* Официант (Имя и фамилия официанта, Номер телефона, Должность, Работает).
* Столик (Номер столика, Название зала, Количество мест).
* Заказ (Номер заказа, Имя официанта, Номер столика, Дата).
* Заказано (Номер заказа, Название блюда).
* Блюдо (Название блюда, Пищевая ценность, Вес).
* Содержит (Название блюда, Имя продукта, Количество продукта).
* Продукт (Имя продукта, На кухне, Единицы измерения).
* Закуплено (Номер контракта, Имя продукта, Название фирмы, Количество).
* Поставщик (Название фирмы).
* Контракт (Номер контракта, Сумма, Дата).

Все отношения отражают объекты или зависимости предметной области, избыточных отношений нет.

* 1. **Проверка соответствия НФБК**
* **Официант** (Имя и фамилия официанта, Номер телефона, Должность, Работает)

**Функциональные зависимости:**

Имя и фамилия официанта → Номер телефона, Должность, Работает.

*Номер телефона → Имя и фамилия официанта, Должность, Работает* – Также является возможным ключом.

Поскольку у двух разных работников может быть одна и та же должность (коллеги), один и тот же статус работы (оба работают или оба уволены), все остальные функциональные зависимости избыточны, так как получаются из приведенных выше путем добавления атрибутов.

Отношение **Официант** находится в НФБК, т.к. все его детерминанты являются возможными ключами.

* **Столик** (Номер столика, Название зала, Количество мест)

**Функциональные зависимости:**

Номер столика → Название зала, Количество мест.

Других функциональных зависимостей нет, так как в одном и том же зале может находиться несколько столиков одинаковой вместимости.

Отношение **Столик** находится в НФБК, т.к. все его детерминанты являются возможными ключами.

* **Заказ** (Номер заказа, Имя официанта, Номер столика, Дата)

**Функциональные зависимости:**

Номер заказа → Имя официанта, Номер столика, Дата.

*Номер столика, Дата → Номер заказа, Имя официанта* – Также является возможным ключом.

*Имя официанта, Дата → Номер заказа, Номер столика* – Также является возможным ключом.

Другие функциональные зависимости избыточны, так как получаются из приведенных выше путем добавления атрибутов.

Отношение **Заказ** находится в НФБК, т.к. все его детерминанты являются возможными ключами.

* **Блюдо** (Название блюда, Пищевая ценность, Вес)

**Функциональные зависимости:**

Название блюда → Пищевая ценность, Вес.

Других функциональных зависимостей нет, так как в меню могут быть два разных блюда с одинаковой пищевой ценностью и весом.

Отношение **Блюдо** находится в НФБК, т.к. все его детерминанты являются возможными ключами.

* **Продукт** (Имя продукта, На кухне, Единицы измерения)

**Функциональные зависимости:**

Имя продукта → На кухне, Единицы измерения.

Других функциональных зависимостей нет, так как могут существовать два разных продукта, измеряющихся в одних и тех же единицах измерения, при этом на кухне бы хранилось равное их количество.

Отношение **Продукт** находится в НФБК, т.к. все его детерминанты являются возможными ключами.

* **Поставщик** (Название фирмы)

Функциональных зависимостей нет, так как отношение содержит всего один, ключевой, атрибут.

Отношение **Поставщик** находится в НФБК, т.к. не имеет функциональных зависимостей.

* **Контракт** (Номер контракта, Сумма, Дата)

**Функциональные зависимости:**

Номер контракта → Сумма, Дата.

Других функциональных зависимостей нет, так как в одно и то же время на одну и ту же сумму могут быть заключены несколько разных контрактов.

Отношение **Контракт** находится в НФБК, т.к. все его детерминанты являются возможными ключами.

* **Заказано** (Номер заказа, Название блюда)

Функциональных зависимостей нет, так как в одном заказе могут присутствовать одни и те же блюда.

Отношение **Заказано** находится в НФБК, т.к. не имеет функциональных зависимостей.

* **Содержит** (Название блюда, Имя продукта, Количество продукта)

**Функциональные зависимости:**

Название блюда, Имя продукта → Количество продукта.

Других функциональных зависимостей нет, так как:

* В одном блюде может использоваться равное количество разных продуктов.
* Одно и то же количество какого-либо продукта может использоваться в разных блюдах.

Отношение **Содержит** находится в НФБК, т.к. все его детерминанты являются возможными ключами.

* **Закуплено** (Номер контракта, Имя продукта, Название фирмы, Количество продукта)

**Функциональные зависимости:**

Номер контракта, Имя продукта, Название фирмы → Количество продукта.

Других функциональных зависимостей нет, так как:

* По одному контракту может быть закуплено одно и то же количество какого-либо продукта у разных поставщиков.
* По одному контракту у одного и того же поставщика может быть закуплено какое-либо количество разных продуктов.
* Один и тот же продукт в определенном количестве может быть закуплен у одного поставщика по разным контрактам.

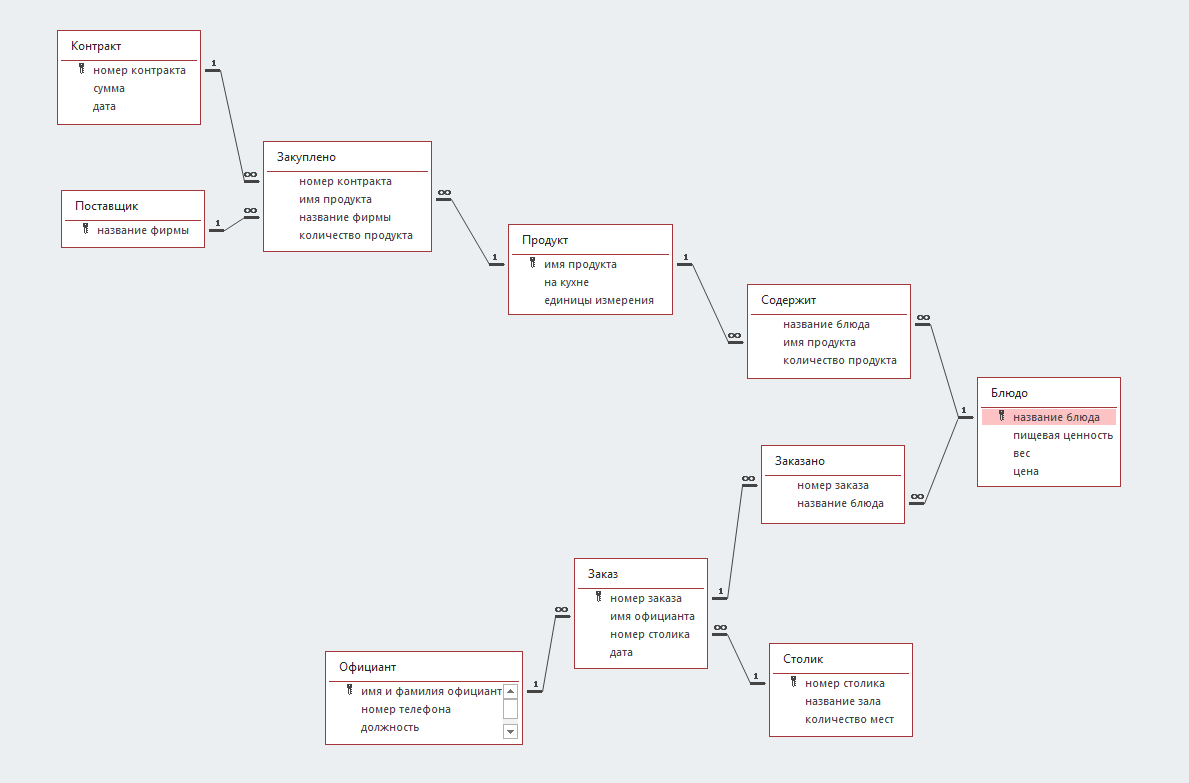
Отношение **Закуплено** находится в НФБК, т.к. все его детерминанты являются возможными ключами.

Таким образом, все отношения находятся в НФБК.

Для всех объектных отношений в качестве ключей выбраны наиболее короткие, не составные, возможные ключи. Необходимости в создании дополнительных ключевых атрибутов для предотвращения избыточного дублирования данных, нет.

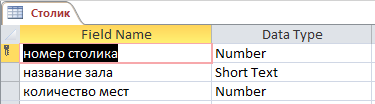
# Структуры таблиц и постоянные связи между таблицами

Постоянные связи между отношениями:

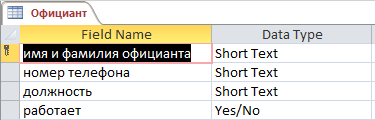


## Структуры таблиц

Ниже представлены структуры всех таблиц спроектированной базы данных.

* Таблица «Столик»

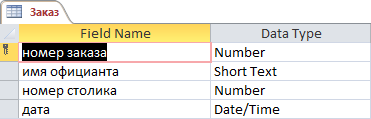
Все атрибуты являются обязательными, пустые строки запрещены.

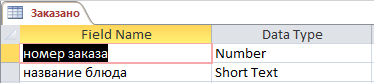
* Таблица «Официант»

Все атрибуты являются обязательными, пустые строки запрещены.

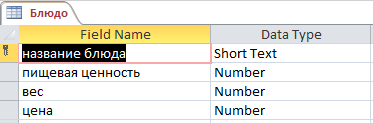
Маска для ввода поля **Номер телефона**: *!\(999") "000\-0000;;\_*

* Таблица «Заказ»

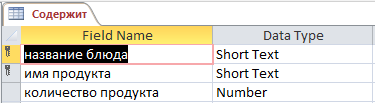
 Все атрибуты являются обязательными, пустые строки запрещены.

* Таблица «Заказано»

Все атрибуты являются обязательными, пустые строки запрещены.

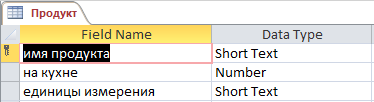
* Таблица «Блюдо»

Все атрибуты являются обязательными, пустые строки запрещены.

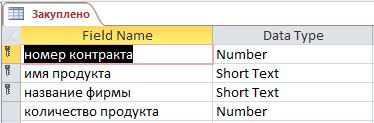
* Таблица «Содержит»

Все атрибуты являются обязательными, пустые строки запрещены. Для увеличения производительности поля **Название блюда** и **Имя продукта** добавлены в уникальный индекс.

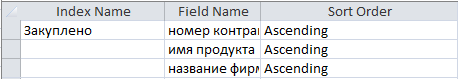


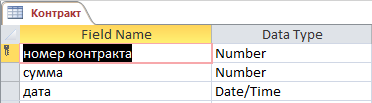
* Таблица «Продукт»

Все атрибуты являются обязательными, пустые строки запрещены.

* Таблица «Закуплено»

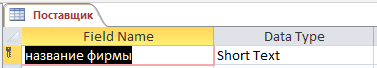
Все атрибуты являются обязательными, пустые строки запрещены. Для увеличения производительности поля **Номер контракта**, **Имя продукта** и **Название фирмы** добавлены в уникальный индекс.



* Таблица «Контракт»

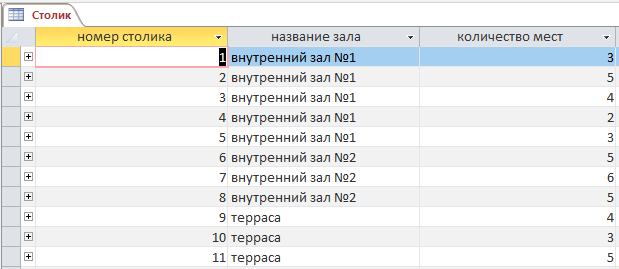
Все строки являются обязательными, пустые строки запрещены.

Маска для ввода поля **Номер контракта**: *#####*

* Таблица «Поставщик»

Все строки являются обязательными, пустые строки запрещены.

# Содержимое таблиц

Таблица «Столик»:

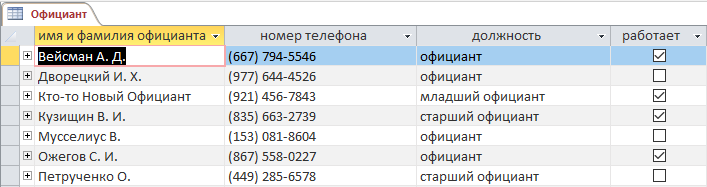
Таблица «Официант»:

Таблица «Заказ»:

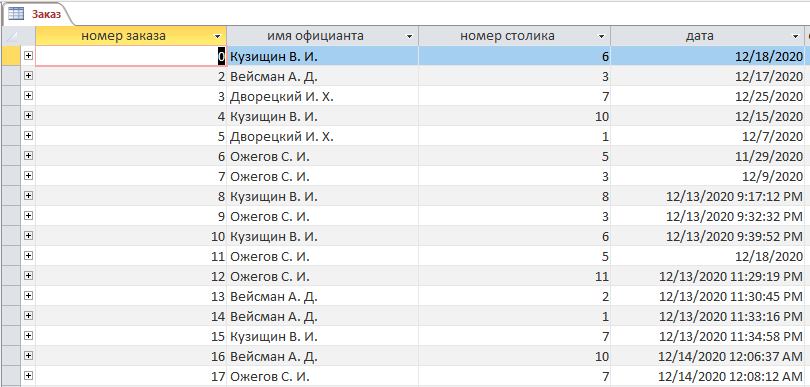


Таблица «Заказано»:

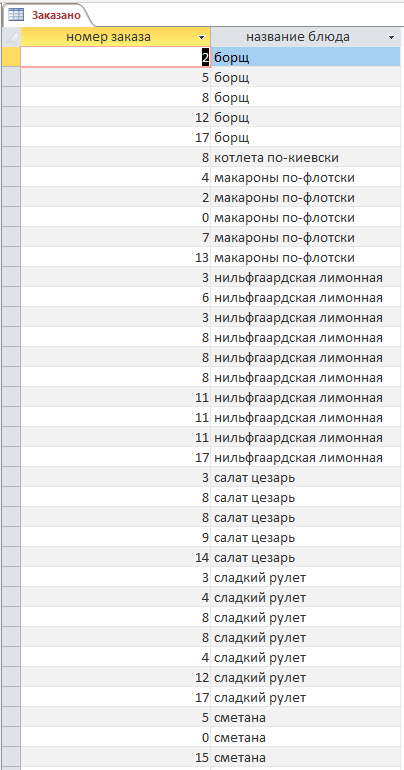


Таблица «Блюдо»:

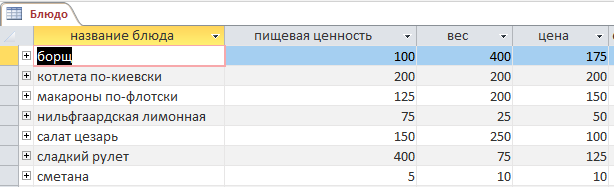


Таблица «Содержит»:

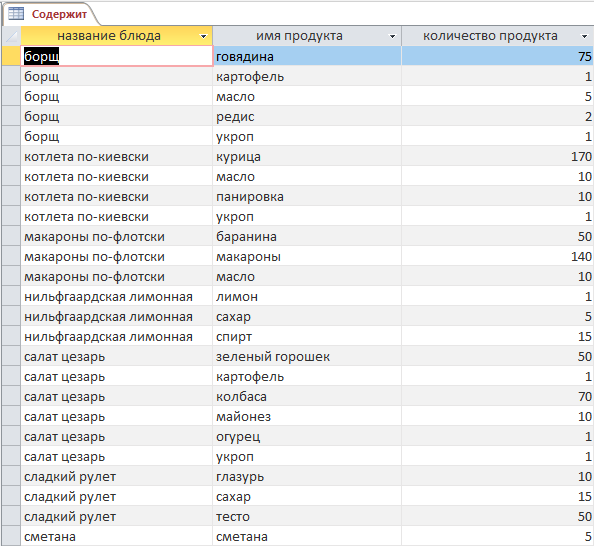
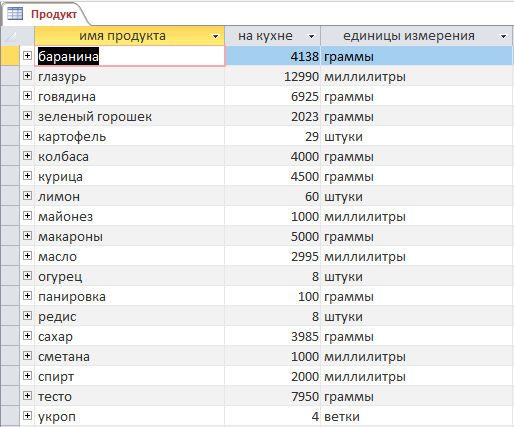
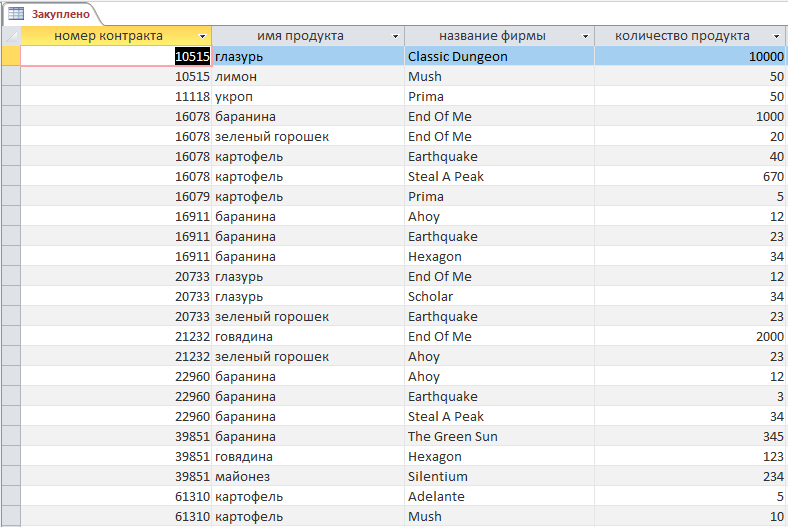


Таблица «Продукт»:



Таблица «Закуплено»:

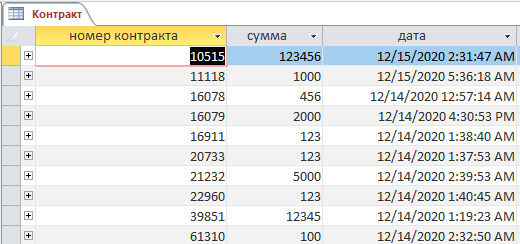
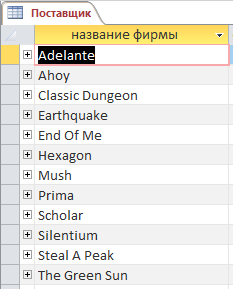
Таблица «Контракт»:

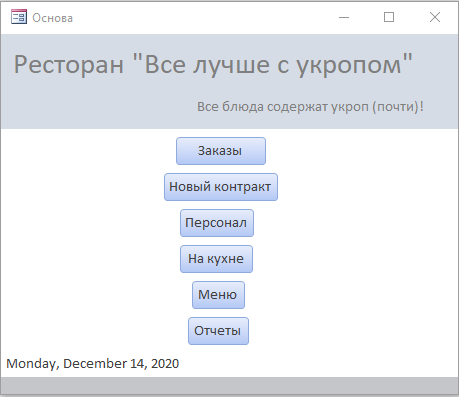
Таблица «Поставщик»:



# Схема иерархии интерфейса

# Описание экранных форм

## Основа

****

***Назначение****:* предоставляет пользователю возможность выбрать интересующую его информацию.

***Кнопки****:*

1. Заказы

По нажатию открывается форма «Заказы».

1. Новый контракт

По нажатию открывается форма «Контракт».

1. Персонал

По нажатию открывается форма «Официанты».

1. На кухне

По нажатию открывается отчет «Блюдо» в режиме просмотра.

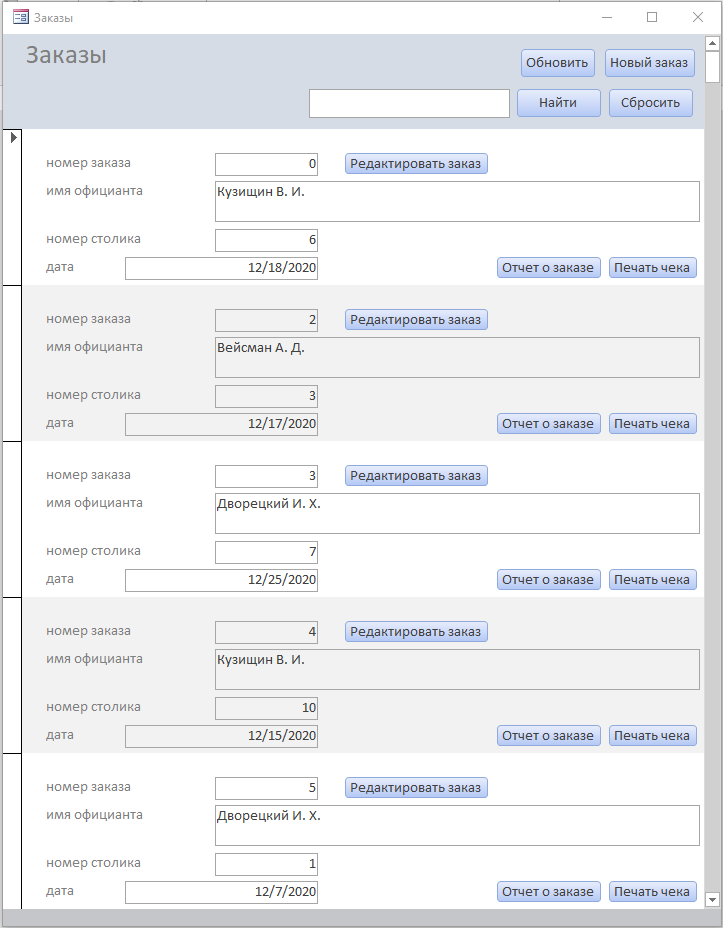
1. Меню

По нажатию открывается отчет «Меню» в режиме просмотра.

1. Отчеты

По нажатию открывается форма «Финансовые отчеты».

## Заказы

****

***Назначение****:* демонстрирует все когда-либо сделанные заказы, осуществляет поиск заказа, позволяет составить отчеты по любому из них.

***Источник******данных****:* таблица **Заказ**.

***Кнопки****:*

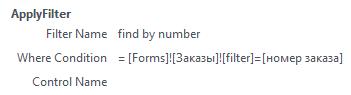
1. Новый заказ

По нажатию открывается форма «Заказ» в режиме добавления данных.

1. Обновить

По нажатию обновляется текущая форма.

1. Найти

По нажатию производится фильтрация по полю **Номер заказа**, выбираются записи, в которых поле равно значению, введенному в поле для ввода **filter**:

1. Сбросить

По нажатию очищается фильтрация.

1. Редактировать заказ

По нажатию открывается форма «Заказ» для соответствующей записи.

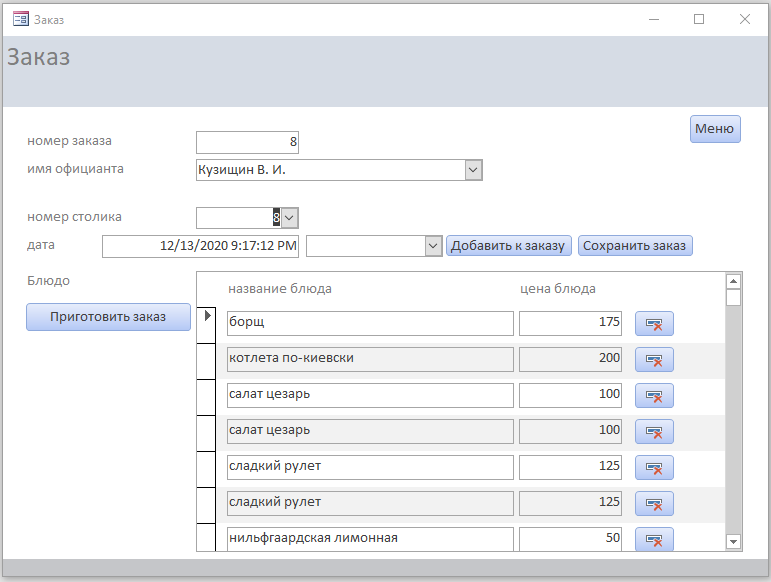
1. Отчет о заказе

По нажатию открывается отчет «Заказ» для соответствующей записи в режиме просмотра.

1. Печать чека

По нажатию открывается отчет «Чек» для соответствующей записи в режиме просмотра.

## Заказ



При добавлении нового заказа номер заказа генерируется автоматически как следующее за максимальным среди существующих номеров заказов число.

В качестве источника данных для поля ввода **название блюда** используется запрос «Проверить доступные блюда». Это сделано для защиты от того, что на очередное блюдо, добавленное в заказ, может не хватить продуктов на кухне.

***Назначение****:* позволяет отредактировать данный заказ или создать новый.

***Источник******данных****:* таблицы **Заказы**, **Заказано**. В качестве источников данных для выпадающих списков **имя официанта**, **номер столика** и **название блюда** выступают таблицы **Официант** и **Столик** и запрос **Получить доступные блюда** соответственно.

***Фильтрация*:** по **Номеру заказа**.

***Кнопки****:*

1. Меню

По нажатию открывается отчет «Меню» в режиме просмотра.

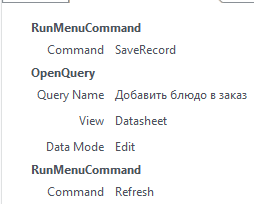
1. Приготовить заказ

По нажатию выполняется запрос «Приготовить заказ».

1. Сохранить заказ

По нажатию сохраняется выбранный заказ.

1. Добавить заказ

****По нажатию выполняется встроенный макрос:

## Официанты

***Назначение****:* демонстрирует всех работающих и когда-либо работавших в ресторане официантов, осуществляет поиск официанта по имени.

***Источник******данных****:* таблица **Официант**.

***Кнопки****:*

1. Новый официант

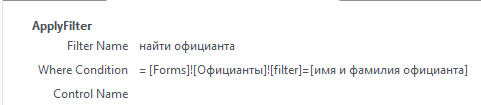
По нажатию открывается форма «Официант» в режиме добавления данных.

1. Обновить

По нажатию обновляется текущая форма.

1. Найти

По нажатию производится фильтрация по полю **Имя и фамилия официанта**, выбираются записи, в которых поле равно значению, введенному в поле для ввода **filter**:



1. Сбросить

По нажатию очищается фильтрация.

1. Открыть анкету

По нажатию открывается форма «Официант» для соответствующей записи.

## Официант

***Назначение****:* позволяет отредактировать личную информацию для существующего официанта или создать для нового.

***Источник******данных****:* таблицы **Официант**, **Заказ**.

***Фильтрация*:** по **Имени и фамилии официанта**.

***Кнопки****:*

1. Сохранить

По нажатию сохраняется выбранный заказ.

## Контракт

Номер нового контракта генерируется автоматически как случайное число.

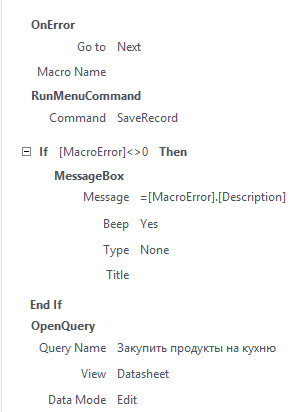
***Назначение****:* позволяет заключить новый контракт, указав закупаемые продукты.

***Источник******данных****:* таблица **Закуплено**.

***Кнопки****:*

1. Заключить

По нажатию выполняется встроенный макрос:



## Финансовые отчеты

По умолчанию в поле ввода **начало** устанавливается дата, на месяц до текущей, а в поле **конец** – текущая.

***Назначение****:* позволяет указать временной промежуток, для которого необходимо продемонстрировать отчет о доходах / расходах ресторана.

***Кнопки****:*

1. Открыть отчет

По нажатию открывается отчет «Финансовые отчеты» для соответствующей записи в режиме просмотра.

# Описание запросов

## Добавить блюдо в заказ

***Формулировка запроса***: добавить блюдо, название которого содержится в поле ввода **Forms![Заказ]![номер заказа]** к заказу, номер которого содержится в поле ввода **Forms![Заказ]![название блюда]**.

***SQL – код запроса***:

INSERT INTO Заказано ( [номер заказа], [название блюда] )

SELECT Forms![Заказ]![номер заказа] AS Expr1, Forms![Заказ]![название блюда] AS Expr2;

Используется в форме **Заказ**, вызывается встроенным макросом по нажатию кнопки **Добавить к заказу**.

## Получить продукты для заказа

***Формулировка запроса***: вывести названия тех продуктов, которые необходимы для приготовления определенного заказа (поле **имя продукта**), необходимое их количество (поле **количество**) и их количество, хранящееся на кухне ресторана (поле **на кухне**).

***SQL – код запроса***:

SELECT Продукт.[имя продукта], Sum(Содержит.[количество продукта]) AS количество, Avg(Продукт.[на кухне]) AS [на кухне]

FROM (Продукт INNER JOIN Содержит ON Продукт.[имя продукта] = Содержит.[имя продукта]) INNER JOIN (Заказано INNER JOIN Блюдо ON Заказано.[название блюда] = Блюдо.[название блюда]) ON Содержит.[название блюда] = Блюдо.[название блюда]

WHERE ((([Forms]![Заказ]![номер заказа])=[номер заказа]))

GROUP BY Продукт.[имя продукта];

***Пример результата выполнения (для заказа номер 8)***:

Используется в запросе **Приготовить заказ** в качестве вложенного запроса.

## Проверить доступные блюда

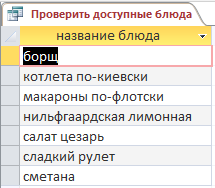
***Формулировка запроса***: вывести названия тех блюд, которые можно приготовить из имеющихся ингредиентов.

***SQL – код запроса***:

SELECT inner\_query.[название блюда]

FROM (SELECT Содержит.[название блюда], Min(Продукт.[на кухне]/Содержит.[количество продукта]) AS количество FROM Блюдо INNER JOIN (Продукт INNER JOIN Содержит ON Продукт.[имя продукта] = Содержит.[имя продукта]) ON Блюдо.[название блюда] = Содержит.[название блюда] GROUP BY Содержит.[название блюда]) AS inner\_query

WHERE (((inner\_query.количество)>1));

***Пример результата выполнения***:

Используется в форме **Заказ** в качестве источника данных для выпадающего списка **название блюда**.

## Закупить продукты на кухню

***Формулировка запроса***: добавить продукты, которые были закуплены по контракту, номер которого содержится в поле ввода **Forms]![Контракт]![номер контракта]** на кухню.

***SQL – код запроса***:

UPDATE Продукт INNER JOIN Закуплено ON Продукт.[имя продукта] = Закуплено.[имя продукта] SET Продукт.[на кухне] = Продукт.[на кухне]+Закуплено.[количество продукта]

WHERE (([Forms]![Контракт]![номер контракта])=[Закуплено].[номер контракта]);

Используется в форме **Контракт**, вызывается встроенным макросом по нажатию кнопки **Заключить**.

## Приготовить заказ

***Формулировка запроса***: убрать с кухни столько таких продуктов, чтобы из них можно было приготовить все блюда определенного заказа, если на кухне достаточно продуктов, и не убирать ничего, если продуктов недостаточно.

***SQL – код запроса***:

UPDATE (Продукт INNER JOIN Содержит ON Продукт.[имя продукта] = Содержит.[имя продукта]) INNER JOIN (Заказано INNER JOIN Блюдо ON Заказано.[название блюда] = Блюдо.[название блюда]) ON Содержит.[название блюда] = Блюдо.[название блюда] SET Продукт.[на кухне] = Продукт.[на кухне]-Содержит.[количество продукта]

WHERE ((([Forms]![Заказ]![номер заказа])=[номер заказа]) AND ('True' IN (SELECT IIf(minimum<0, 'False', 'True') AS mlz FROM (SELECT MIN([на кухне] - [количество]) AS minimum FROM [Получить продукты для заказа]))));

Используется в форме **Заказ**, вызывается встроенным макросом по нажатию кнопки **Приготовить заказ**.

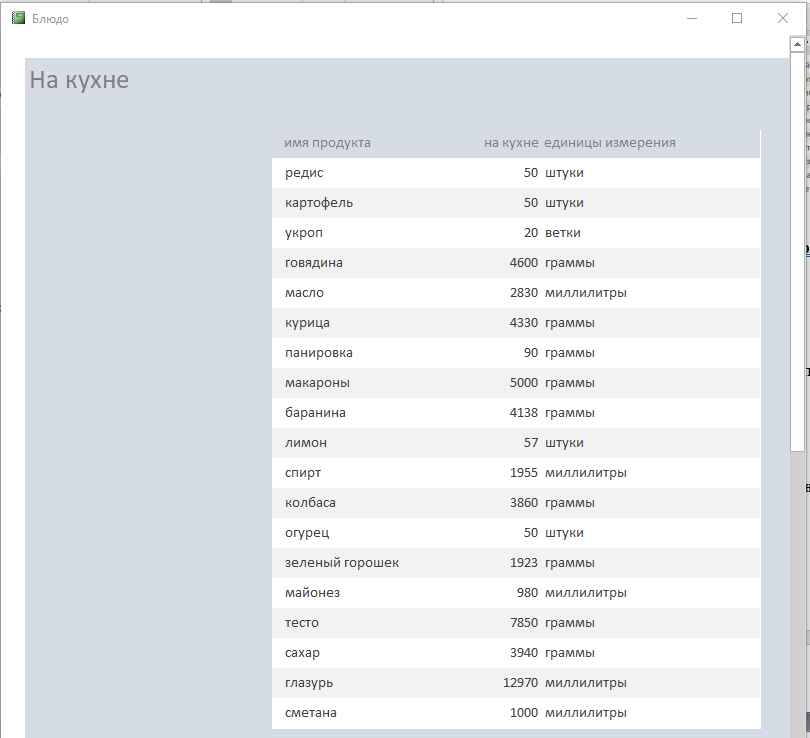
# Описание отчетов

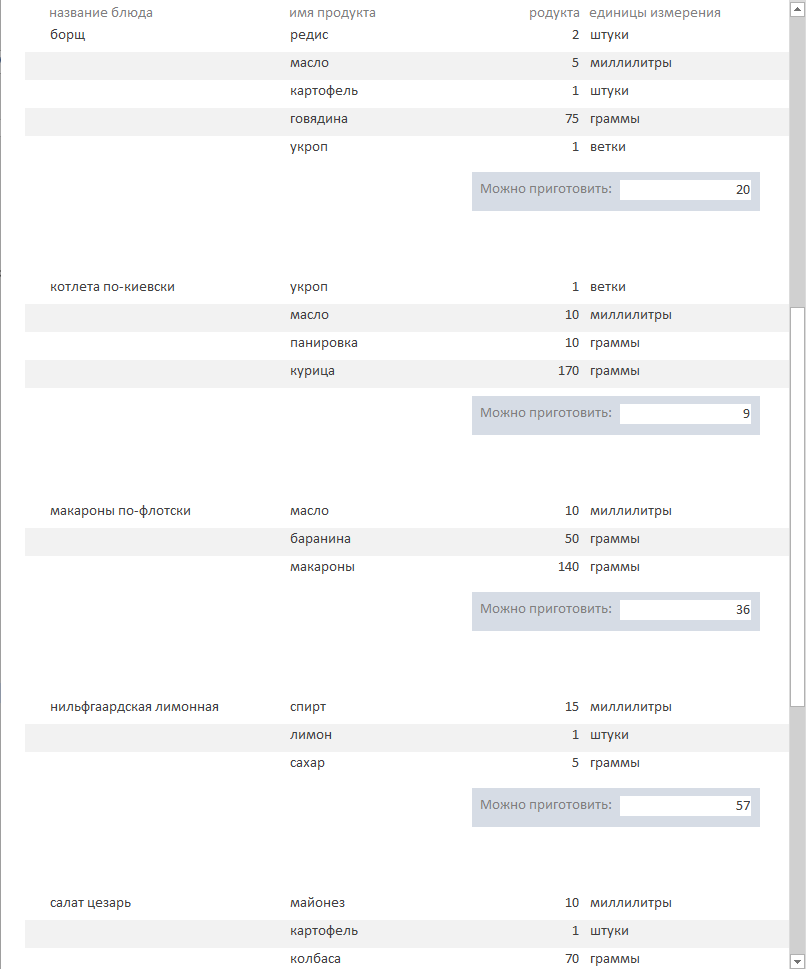
## Меню

***Источник данных*** – таблицы **Блюдо**, **Содержит** и **Продукт**.

***Сортировка***: название блюда **–** по алфавиту.

## Блюдо





***Источник данных*** – таблицы **Блюдо**, **Содержит**, **Продукт**, а также встроенный SQL запрос для встроенного отчета о количестве блюд, которые можно приготовить из имеющихся ингредиентов:

SELECT Содержит.[название блюда], Продукт.[на кухне]/Содержит.[количество продукта] AS количество

FROM Блюдо INNER JOIN (Продукт INNER JOIN Содержит ON Продукт.[имя продукта] = Содержит.[имя продукта]) ON Блюдо.[название блюда] = Содержит.[название блюда];

***Сортировка***: название блюда – по алфавиту.

## Финансовые отчеты

***Источник данных*** – встроенные SQL запросы для встроенных отчетов «Доходы» м «Расходы»:

* Доходы:

SELECT Заказ.[номер заказа], Заказ.дата, Блюдо.[название блюда], Блюдо.цена, Заказ.дата, Заказ.дата

FROM Блюдо INNER JOIN (Заказ INNER JOIN Заказано ON Заказ.[номер заказа] = Заказано.[номер заказа]) ON Блюдо.[название блюда] = Заказано.[название блюда]

WHERE (((Заказ.дата)>=[Forms]![Финансовые отчеты]![начало] And (Заказ.дата)<=[Forms]![Финансовые отчеты]![конец]));

* Расходы:

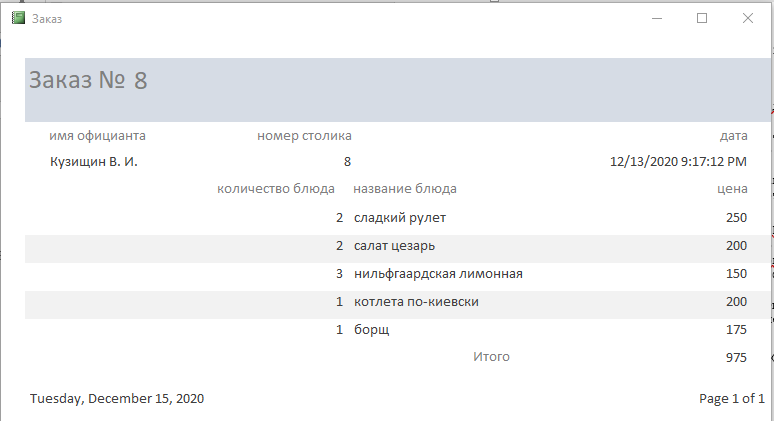
SELECT Контракт.[номер контракта], Контракт.сумма, Контракт.дата, Закуплено.[имя продукта], Закуплено.[название фирмы], Закуплено.[количество продукта], Контракт.дата, Контракт.дата

FROM Контракт INNER JOIN Закуплено ON Контракт.[номер контракта] = Закуплено.[номер контракта]

WHERE (((Контракт.дата)>=[Forms]![Финансовые отчеты]![начало] And (Контракт.дата)<=[Forms]![Финансовые отчеты]![конец]));

***Сортировка***: номера заказов в отчете «Доходы» и номера контрактов в отчете «Расходы» **–** по возрастанию.

## Заказ



***Источник данных*** – встроенный SQL запрос:

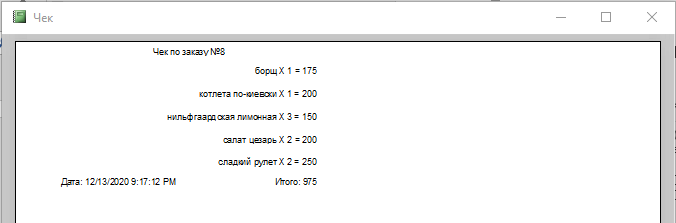
SELECT First(Заказ.[номер заказа]) AS [номер заказа], First(Заказ.[имя официанта]) AS [имя официанта], First(Заказ.[номер столика]) AS [номер столика], First(Заказ.дата) AS дата, Блюдо.[название блюда], Sum(Блюдо.цена) AS цена, Count(Блюдо.[название блюда]) AS количество

FROM Блюдо INNER JOIN (Заказ INNER JOIN Заказано ON Заказ.[номер заказа] = Заказано.[номер заказа]) ON Блюдо.[название блюда] = Заказано.[название блюда]

WHERE ((([Forms]![Заказы]![номер заказа])=[Заказ].[номер заказа]))

GROUP BY Блюдо.[название блюда];

## Чек



Источник данных – встроенный SQL запрос:

SELECT First(Заказ.[номер заказа]) AS [номер заказа], First(Заказ.[имя официанта]) AS [имя официанта], First(Заказ.[номер столика]) AS [номер столика], First(Заказ.дата) AS дата, Блюдо.[название блюда], Sum(Блюдо.цена) AS цена, Count(Блюдо.[название блюда]) AS количество

FROM Блюдо INNER JOIN (Заказ INNER JOIN Заказано ON Заказ.[номер заказа] = Заказано.[номер заказа]) ON Блюдо.[название блюда] = Заказано.[название блюда]

WHERE ((([Forms]![Заказы]![номер заказа])=[Заказ].[номер заказа]))

GROUP BY Блюдо.[название блюда];

# **Заключение**

В ходе выполнения данного индивидуального задания была реализована программная система для функционирования ресторана.

Был реализован поиск и изменение информации о всех когда-либо сделанных заказах и официантах.

Была добавлена возможность вносить в базу данных информацию о контрактах, заключенных рестораном с поставщиками продуктов, из которых состоят блюда, а также автоматическое добавление на кухню ресторана всех продуктов, содержащихся в контракте и удаление продуктов, составляющих блюда в новом заказе. При этом было уделено внимание тому, чтобы приготовить можно было продуктов не больше, чем возможно из находящихся на кухне продуктов.

О всех доходах и расходах ресторана, а также о состоянии кухни были составлены подробные отчеты.

Поиск и операции данных были описаны при помощи запросов на языке SQL, как глобальных, так и встроенных в соответствующие элементы

Таблицы, запросы, формы и отчеты реализованы с помощью QBE, макросов и соответствующих мастеров.