

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
1.1. Задание.....	3
2. Проектирование БД.....	5
2.1. Описание сущностей	5
2.2. ER-модель	5
2.3. Реляционная модель	7
2.4. Проверка соответствия НФБК	9
3. Структуры таблиц и постоянные связи между таблицами	14
3.1. Структуры таблиц	14
4. Содержимое таблиц	17
5. Схема иерархии интерфейса	22
6. Описание экранных форм.....	23
6.1. Основа	23
6.2. Заказы.....	24
6.3. Заказ	25
6.4. Официанты	28
6.5. Официант	29
6.6. Контракт.....	30
6.7. Финансовые отчеты	31
7. Описание запросов.....	32
7.1. Добавить блюдо в заказ.....	32
7.2. Получить продукты для заказа	32
7.3. Проверить доступные блюда	33
7.4. Закупить продукты на кухню	34
7.5. Приготовить заказ.....	34
8. Описание отчетов.....	35
8.1. Меню	35
8.2. Блюдо	35
8.3. Финансовые отчеты	38
8.4. Заказ	39
8.5. Чек	40
Заключение.....	41

1. Постановка задачи

1.1. 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 поставке 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 заведении
- 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

2. 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.

2.1. Описание сущностей

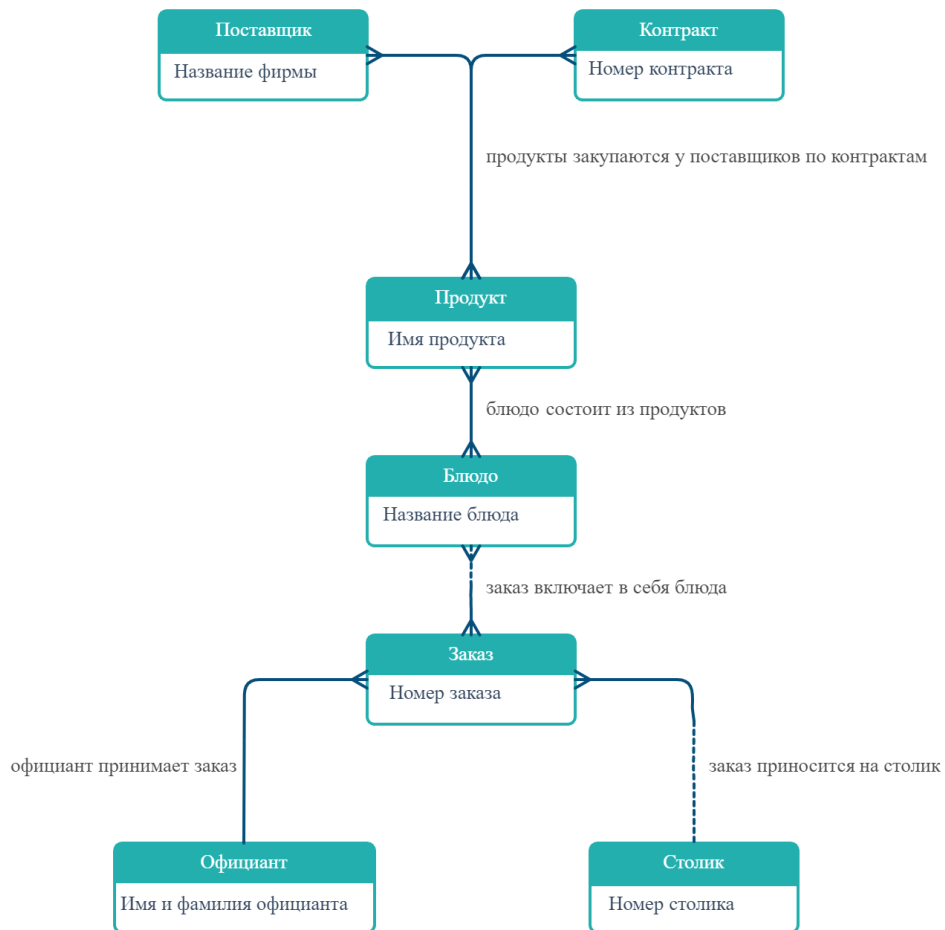
Можно выделить следующие сущности:

- «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».

2.2. 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.

2.3. 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.

2.4. 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.

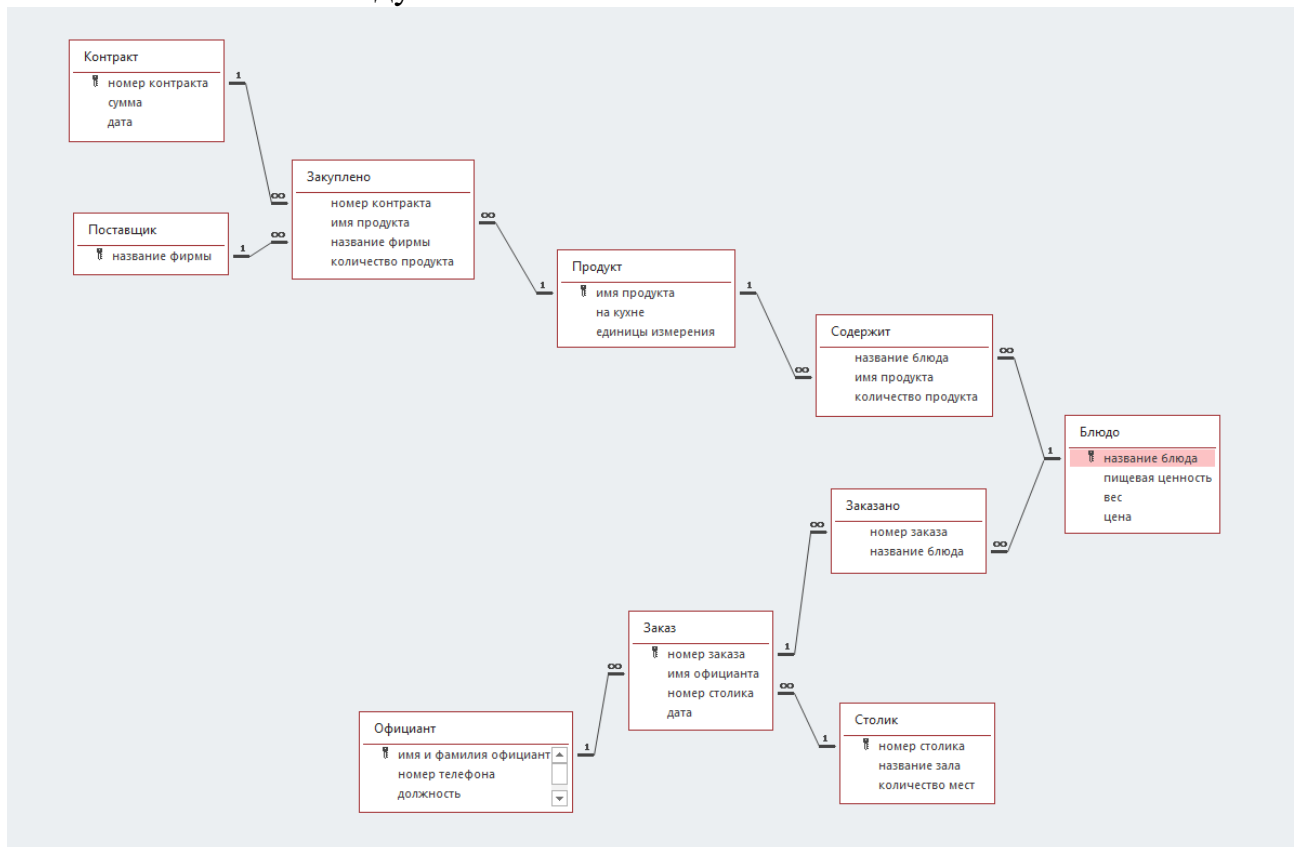
Отношение **Закуплено** находится в НФБК, т.к. все его детерминанты являются возможными ключами.

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.

3. Структуры таблиц и постоянные связи между таблицами

Постоянные связи между отношениями:



3.1. Структуры таблиц

Ниже представлены структуры всех таблиц спроектированной базы данных.

- Таблица «Столик»

Столик	
Field Name	Data Type
номер столика	Number
название зала	Short Text
количество мест	Number

Все атрибуты являются обязательными, пустые строки запрещены.

- Таблица «Официант»

Официант	
Field Name	Data Type
имя и фамилия официанта	Short Text
номер телефона	Short Text
должность	Short Text
работает	Yes/No

Все атрибуты являются обязательными, пустые строки запрещены.

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

- Таблица «Заказ»

Заказ	
Field Name	Data Type
номер заказа	Number
имя официанта	Short Text
номер столика	Number
дата	Date/Time

Все атрибуты являются обязательными, пустые строки запрещены.

- Таблица «Заказано»

Заказано	
Field Name	Data Type
номер заказа	Number
название блюда	Short Text

Все атрибуты являются обязательными, пустые строки запрещены.

- Таблица «Блюдо»

Блюдо	
Field Name	Data Type
название блюда	Short Text
пищевая ценность	Number
вес	Number
цена	Number

Все атрибуты являются обязательными, пустые строки запрещены.

- Таблица «Содержит»

Содержит	
Field Name	Data Type
название блюда	Short Text
имя продукта	Short Text
количество продукта	Number

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

Index Name	Field Name	Sort Order
Содержит	название блю,	Ascending
	имя продукта	Ascending

- Таблица «Продукт»

Продукт		
	Field Name	Data Type
имя продукта		Short Text
на кухне		Number
единицы измерения		Short Text

Все атрибуты являются обязательными, пустые строки запрещены.

- Таблица «Закуплено»

Закуплено		
	Field Name	Data Type
номер контракта		Number
имя продукта		Short Text
название фирмы		Short Text
количество продукта		Number

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

Index Name	Field Name	Sort Order
Закуплено	номер контракта	Ascending
	имя продукта	Ascending
	название фирмы	Ascending

- Таблица «Контракт»

Контракт		
	Field Name	Data Type
номер контракта		Number
сумма		Number
дата		Date/Time

Все строки являются обязательными, пустые строки запрещены.

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

- Таблица «Поставщик»

Поставщик		
	Field Name	Data Type
название фирмы		Short Text

Все строки являются обязательными, пустые строки запрещены.

4. Содержимое таблиц

Таблица «Столик»:

Столик			
	номер столика	название зала	количество мест
+	1	внутренний зал №1	3
+	2	внутренний зал №1	5
+	3	внутренний зал №1	4
+	4	внутренний зал №1	2
+	5	внутренний зал №1	3
+	6	внутренний зал №2	5
+	7	внутренний зал №2	6
+	8	внутренний зал №2	5
+	9	терраса	4
+	10	терраса	3
+	11	терраса	5

Таблица «Официант»:

Официант				
	имя и фамилия официанта	номер телефона	должность	работает
+	Вейсман А. Д.	(667) 794-5546	официант	<input checked="" type="checkbox"/>
+	Дворецкий И. Х.	(977) 644-4526	официант	<input type="checkbox"/>
+	Кто-то Новый Официант	(921) 456-7843	младший официант	<input checked="" type="checkbox"/>
+	Кузищин В. И.	(835) 663-2739	старший официант	<input checked="" type="checkbox"/>
+	Мусселиус В.	(153) 081-8604	официант	<input type="checkbox"/>
+	Ожегов С. И.	(867) 558-0227	официант	<input checked="" type="checkbox"/>
+	Петрученко О.	(449) 285-6578	старший официант	<input type="checkbox"/>

Таблица «Заказ»:

Заказ				
	номер заказа	имя официанта	номер столика	дата
+	0	Кузищин В. И.	6	12/18/2020
+	2	Вейсман А. Д.	3	12/17/2020
+	3	Дворецкий И. Х.	7	12/25/2020
+	4	Кузищин В. И.	10	12/15/2020
+	5	Дворецкий И. Х.	1	12/7/2020
+	6	Ожегов С. И.	5	11/29/2020
+	7	Ожегов С. И.	3	12/9/2020
+	8	Кузищин В. И.	8	12/13/2020 9:17:12 PM
+	9	Ожегов С. И.	3	12/13/2020 9:32:32 PM
+	10	Кузищин В. И.	6	12/13/2020 9:39:52 PM
+	11	Ожегов С. И.	5	12/18/2020
+	12	Ожегов С. И.	11	12/13/2020 11:29:19 PM
+	13	Вейсман А. Д.	2	12/13/2020 11:30:45 PM
+	14	Вейсман А. Д.	1	12/13/2020 11:33:16 PM
+	15	Кузищин В. И.	7	12/13/2020 11:34:58 PM
+	16	Вейсман А. Д.	10	12/14/2020 12:06:37 AM
+	17	Ожегов С. И.	7	12/14/2020 12:08:12 AM

Таблица «Заказано»:

Заказано	
номер заказа	название блюда
2	борщ
5	борщ
8	борщ
12	борщ
17	борщ
8	котлета по-киевски
4	макароны по-флотски
2	макароны по-флотски
0	макароны по-флотски
7	макароны по-флотски
13	макароны по-флотски
3	нильфаардская лимонная
6	нильфаардская лимонная
3	нильфаардская лимонная
8	нильфаардская лимонная
8	нильфаардская лимонная
8	нильфаардская лимонная
11	нильфаардская лимонная
11	нильфаардская лимонная
11	нильфаардская лимонная
17	нильфаардская лимонная
3	салат цезарь
8	салат цезарь
8	салат цезарь
9	салат цезарь
14	салат цезарь
3	сладкий рулет
4	сладкий рулет
8	сладкий рулет
8	сладкий рулет
4	сладкий рулет
12	сладкий рулет
17	сладкий рулет
5	сметана
0	сметана
15	сметана

Таблица «Блюдо»:

Блюдо				
	название блюда	пищевая ценность	вес	цена
+	борщ	100	400	175
+	котлета по-киевски	200	200	200
+	макароны по-флотски	125	200	150
+	нильфгаардская лимонная	75	25	50
+	салат цезарь	150	250	100
+	сладкий рулет	400	75	125
+	сметана	5	10	10

Таблица «Содержит»:

Содержит			
	название блюда	имя продукта	количество продукта
	борщ	говядина	75
	борщ	картофель	1
	борщ	масло	5
	борщ	редис	2
	борщ	укроп	1
	котлета по-киевски	курица	170
	котлета по-киевски	масло	10
	котлета по-киевски	панировка	10
	котлета по-киевски	укроп	1
	макароны по-флотски	баранина	50
	макароны по-флотски	макароны	140
	макароны по-флотски	масло	10
	нильфгаардская лимонная	лимон	1
	нильфгаардская лимонная	сахар	5
	нильфгаардская лимонная	спирт	15
	салат цезарь	зеленый горошек	50
	салат цезарь	картофель	1
	салат цезарь	колбаса	70
	салат цезарь	майонез	10
	салат цезарь	огурец	1
	салат цезарь	укроп	1
	сладкий рулет	глазурь	10
	сладкий рулет	сахар	15
	сладкий рулет	тесто	50
	сметана	сметана	5

Таблица «Продукт»:

Продукт			
	имя продукта	на кухне	единицы измерения
+	баранина	4138	граммы
+	глазурь	12990	миллилитры
+	говядина	6925	граммы
+	зеленый горошек	2023	граммы
+	картофель	29	штуки
+	колбаса	4000	граммы
+	курица	4500	граммы
+	лимон	60	штуки
+	майонез	1000	миллилитры
+	макароны	5000	граммы
+	масло	2995	миллилитры
+	огурец	8	штуки
+	панировка	100	граммы
+	редис	8	штуки
+	сахар	3985	граммы
+	сметана	1000	миллилитры
+	спирт	2000	миллилитры
+	тесто	7950	граммы
+	укроп	4	ветки

Таблица «Закуплено»:

Закуплено				
	номер контракта	имя продукта	название фирмы	количество продукта
	10515	глазурь	Classic Dungeon	10000
	10515	лимон	Mush	50
	11118	укроп	Prima	50
	16078	баранина	End Of Me	1000
	16078	зеленый горошек	End Of Me	20
	16078	картофель	Earthquake	40
	16078	картофель	Steal A Peak	670
	16079	картофель	Prima	5
	16911	баранина	Ahoy	12
	16911	баранина	Earthquake	23
	16911	баранина	Hexagon	34
	20733	глазурь	End Of Me	12
	20733	глазурь	Scholar	34
	20733	зеленый горошек	Earthquake	23
	21232	говядина	End Of Me	2000
	21232	зеленый горошек	Ahoy	23
	22960	баранина	Ahoy	12
	22960	баранина	Earthquake	3
	22960	баранина	Steal A Peak	34
	39851	баранина	The Green Sun	345
	39851	говядина	Hexagon	123
	39851	майонез	Silentium	234
	61310	картофель	Adelante	5
	61310	картофель	Mush	10

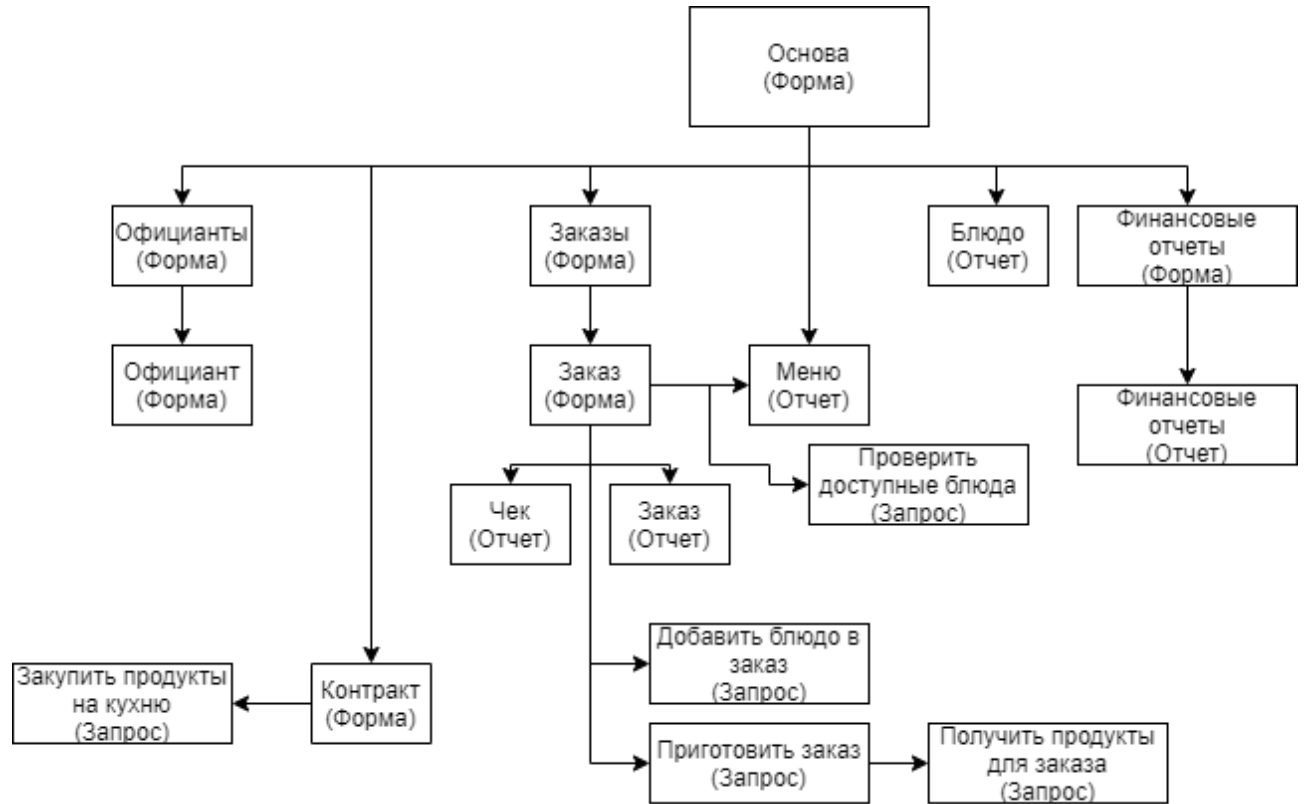
Таблица «Контракт»:

Контракт			
	номер контракта	сумма	дата
+	10515	123456	12/15/2020 2:31:47 AM
+	11118	1000	12/15/2020 5:36:18 AM
+	16078	456	12/14/2020 12:57:14 AM
+	16079	2000	12/14/2020 4:30:53 PM
+	16911	123	12/14/2020 1:38:40 AM
+	20733	123	12/14/2020 1:37:53 AM
+	21232	5000	12/14/2020 2:39:53 AM
+	22960	123	12/14/2020 1:40:45 AM
+	39851	12345	12/14/2020 1:19:23 AM
+	61310	100	12/14/2020 2:32:50 AM

Таблица «Поставщик»:

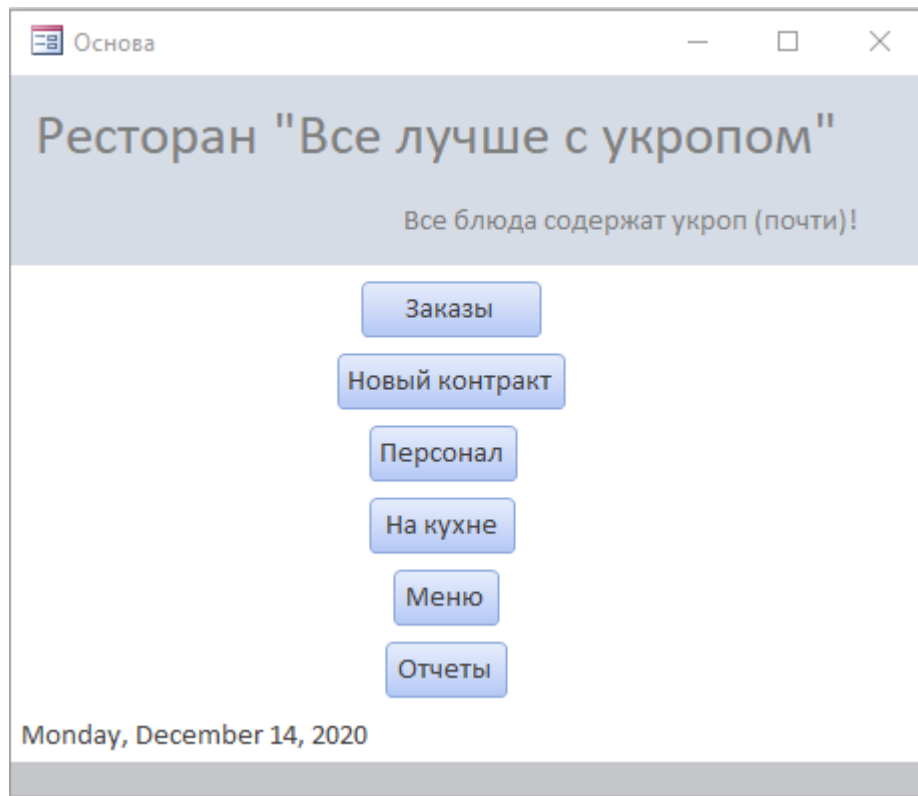
Поставщик	
	название фирмы
+	Adelante
+	Ahoy
+	Classic Dungeon
+	Earthquake
+	End Of Me
+	Hexagon
+	Mush
+	Prima
+	Scholar
+	Silentium
+	Steal A Peak
+	The Green Sun

5. Схема иерархии интерфейса



6. Описание экранных форм

6.1. Основа



Назначение: предоставляет пользователю возможность выбрать интересующую его информацию.

Кнопки:

1.) Заказы

По нажатию открывается форма «Заказы».

2.) Новый контракт

По нажатию открывается форма «Контракт».

3.) Персонал

По нажатию открывается форма «Официанты».

4.) На кухне

По нажатию открывается отчет «Блюда» в режиме просмотра.

5.) Меню

По нажатию открывается отчет «Меню» в режиме просмотра.

6.) Отчеты

По нажатию открывается форма «Финансовые отчеты».

6.2. Заказы

Заказы

ОбновитьНовый заказ

НайтиСбросить

номер заказа	<input type="text" value="0"/>	Редактировать заказ
имя официанта	<input type="text" value="Кузищин В. И."/>	
номер столика	<input type="text" value="6"/>	
дата	<input type="text" value="12/18/2020"/>	Отчет о заказеПечать чека
номер заказа	<input type="text" value="2"/>	Редактировать заказ
имя официанта	<input type="text" value="Вейсман А. Д."/>	
номер столика	<input type="text" value="3"/>	
дата	<input type="text" value="12/17/2020"/>	Отчет о заказеПечать чека
номер заказа	<input type="text" value="3"/>	Редактировать заказ
имя официанта	<input type="text" value="Дворецкий И. Х."/>	
номер столика	<input type="text" value="7"/>	
дата	<input type="text" value="12/25/2020"/>	Отчет о заказеПечать чека
номер заказа	<input type="text" value="4"/>	Редактировать заказ
имя официанта	<input type="text" value="Кузищин В. И."/>	
номер столика	<input type="text" value="10"/>	
дата	<input type="text" value="12/15/2020"/>	Отчет о заказеПечать чека
номер заказа	<input type="text" value="5"/>	Редактировать заказ
имя официанта	<input type="text" value="Дворецкий И. Х."/>	
номер столика	<input type="text" value="1"/>	
дата	<input type="text" value="12/7/2020"/>	Отчет о заказеПечать чека

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.

2.) Обновить

По нажатию обновляется текущая форма.

3.) Найти

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

```
ApplyFilter
  Filter Name  find by number
  Where Condition  = [Forms]![Заказы]![filter]=[номер заказа]
  Control Name
```

4.) Сбросить

По нажатию очищается фильтрация.

5.) Редактировать заказ

По нажатию открывается форма «Заказ» для соответствующей записи.

6.) Отчет о заказе

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

7.) Печать чека

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

6.3. Заказ

Заказ

номер заказа:

имя официанта:

номер столика:

дата:

Блюдо

название блюда	цена блюда	
борщ	175	<input type="button" value="✕"/>
котлета по-киевски	200	<input type="button" value="✕"/>
салат цезарь	100	<input type="button" value="✕"/>
салат цезарь	100	<input type="button" value="✕"/>
сладкий рулет	125	<input type="button" value="✕"/>
сладкий рулет	125	<input type="button" value="✕"/>
нильфгаардская лимонная	50	<input type="button" value="✕"/>

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.

2.) Приготовить заказ

По нажатию выполняется запрос «Приготовить заказ».

3.) Сохранить заказ

По нажатию сохраняется выбранный заказ.

4.) Добавить заказ

По нажатию выполняется встроенный макрос:

RunMenuCommand

Command SaveRecord

OpenQuery

Query Name Добавить блюдо в заказ

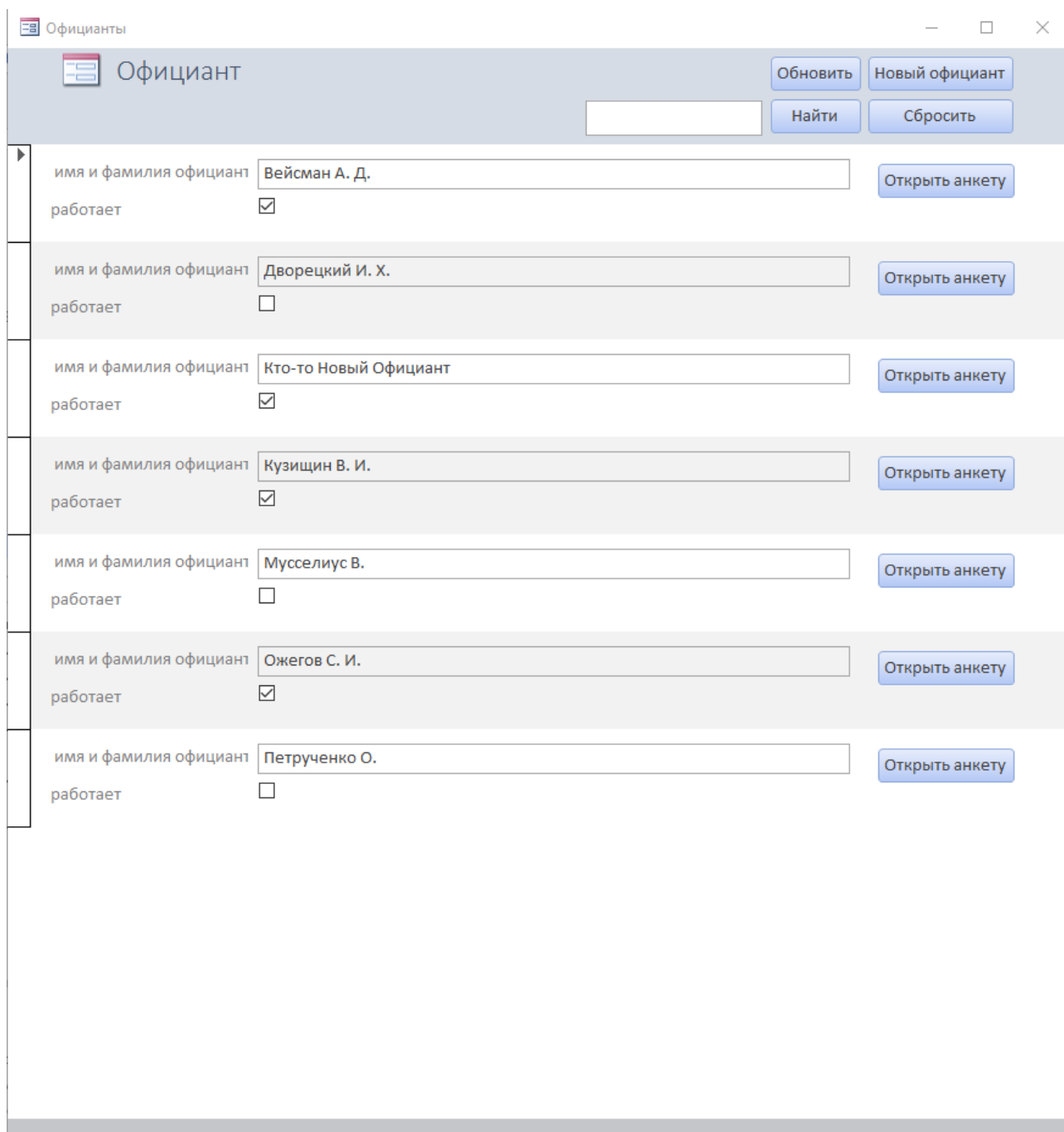
View Datasheet

Data Mode Edit

RunMenuCommand

Command Refresh

6.4. Официанты



Официант		
имя и фамилия официант	Вейсман А. Д.	Открыть анкету
работает	<input checked="" type="checkbox"/>	
имя и фамилия официант	Дворецкий И. Х.	Открыть анкету
работает	<input type="checkbox"/>	
имя и фамилия официант	Кто-то Новый Официант	Открыть анкету
работает	<input checked="" type="checkbox"/>	
имя и фамилия официант	Кузищин В. И.	Открыть анкету
работает	<input checked="" type="checkbox"/>	
имя и фамилия официант	Мусселиус В.	Открыть анкету
работает	<input type="checkbox"/>	
имя и фамилия официант	Ожегов С. И.	Открыть анкету
работает	<input checked="" type="checkbox"/>	
имя и фамилия официант	Петрученко О.	Открыть анкету
работает	<input type="checkbox"/>	

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.

2.) Обновить

По нажатию обновляется текущая форма.

3.) Найти

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

```
ApplyFilter
  Filter Name   найти официанта
  Where Condition = [Forms]![Официанты]![filter]=[имя и фамилия официанта]
  Control Name
```

4.) Сбросить

По нажатию очищается фильтрация.

5.) Открыть анкету

По нажатию открывается форма «Официант» для соответствующей записи.

6.5. Официант

Официант

имя и фамилия официанта: Вейсман А. Д.

номер телефона: (667) 794-5546

должность: официант

работает: ☒

Обслуживание: Заказов принял: 6

Сохранить

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.

6.6. Контракт

The screenshot shows a window titled 'Контракт' (Contract) with a subtitle 'Новый контракт' (New Contract). The form contains the following fields and controls:

- номер контракта** (contract number): A text input field containing the value '63321'.
- сумма** (sum): A text input field containing the value '0'.
- дата** (date): A text input field containing the value '12/23/2020 6:09:19 AM'.
- Заклучить** (Sign): A blue button located to the right of the contract number field.
- Закуплено** (Purchased): A section containing a table with the following headers: 'имя продукта' (product name), 'название фирмы' (company name), and 'количество продукта' (quantity of product).

The table has one row with the following values: an empty product name field, an empty company name field, and a quantity of '0'. There is a small blue button with a trash icon to the right of the quantity field.

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:

```

OnError
    Go to Next
    Macro Name
RunMenuCommand
    Command SaveRecord

☐ If [MacroError]<>0 Then
    MessageBox
        Message =[MacroError].[Description]
        Beep Yes
        Type None
        Title

End If
OpenQuery
    Query Name Закупить продукты на кухню
    View Datasheet
    Data Mode Edit

```

6.7. 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.

7. Описание запросов

7.1. 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**.

7.2. 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):

Получить продукты для заказа		
имя продукта	количество	на кухне
глазурь	20	33093
говядина	75	4525
зеленый горошек	100	1923
картофель	3	49
колбаса	140	3860
курица	170	4330
лимон	3	182
майонез	20	2480
масло	15	2825
огурец	2	50
панировка	10	90
редис	2	48
сахар	45	3940
спирт	45	1000
тесто	100	7850
укроп	4	19

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

7.3. 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:

Проверить доступные блюда	
название блюда	
борщ	
котлета по-киевски	
макароны по-флотски	
нильфгаардская лимонная	
салат цезарь	
сладкий рулет	
сметана	

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

7.4. 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**.

7.5. 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**.

8. Описание отчетов

8.1. Меню

Меню						
название блюда	пищевая ценность	вес	цена	имя продукта	количество продукта	единицы измерения
борщ	100	400	175	Состоит из:		
				редис	2	штуки
				масло	5	миллилитры
				картофель	1	штуки
				говядина	75	граммы
				укроп	1	ветки
котлета по-киевски	200	200	200	Состоит из:		
				укроп	1	ветки
				масло	10	миллилитры
				панировка	10	граммы
				курица	170	граммы
макароны по-флотски	125		150	Состоит из:		
				масло	10	миллилитры
				баранина	50	граммы
				макароны	140	граммы
нильфгаардская лимонная	75	25	50	Состоит из:		
				спирт	15	миллилитры
				лимон	1	штуки
				сахар	5	граммы
салат цезарь	150	250	100	Состоит из:		
				майонез	10	миллилитры
				картофель	1	штуки
				колбаса	70	граммы
				огурец	1	штуки
				укроп	1	ветки
				зеленый горошек	50	граммы
сладкий рулет	400	75	125	Состоит из:		
				тесто	50	граммы
				глазурь	10	миллилитры
				сахар	15	граммы
сметана	5	10	10	Состоит из:		
				сметана	5	миллилитры
Wednesday, December 23, 2020						
Page 1 of 1						

Источник данных – таблицы **Блюдо**, **Содержит** и **Продукт**.
Сортировка: название блюда – по алфавиту.

8.2. Блюдо

На кухне	
имя продукта	на кухне единицы измерения
редис	50 штуки
картофель	50 штуки
укроп	20 ветки
говядина	4600 граммы
масло	2830 миллилитры
курица	4330 граммы
панировка	90 граммы
макароны	5000 граммы
баранина	4138 граммы
лимон	57 штуки
спирт	1955 миллилитры
колбаса	3860 граммы
огурец	50 штуки
зеленый горошек	1923 граммы
майонез	980 миллилитры
тесто	7850 граммы
сахар	3940 граммы
глазурь	12970 миллилитры
сметана	1000 миллилитры

название блюда	имя продукта	количество	единицы измерения
борщ	редис	2	штуки
	масло	5	миллилитры
	картофель	1	штуки
	говядина	75	граммы
	укроп	1	ветки
		Можно приготовить:	20
котлета по-киевски	укроп	1	ветки
	масло	10	миллилитры
	панировка	10	граммы
	курица	170	граммы
		Можно приготовить:	9
макароны по-флотски	масло	10	миллилитры
	баранина	50	граммы
	макароны	140	граммы
		Можно приготовить:	36
нильсгаардская лимонная	спирт	15	миллилитры
	лимон	1	штуки
	сахар	5	граммы
		Можно приготовить:	57
салат цезарь	майонез	10	миллилитры
	картофель	1	штуки
	колбаса	70	граммы

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.

8.3. Финансовые отчеты

Финансовый отчет с 12/15/2020 по 12/16/2020			
Доходы			
номер заказа	дата	название блюда	цена
13	12/15/2020 11:30:45 PM	макароны по-флотски	150
		Итого:	150
17	12/15/2020 12:08:12 AM	нильфгаардская лимонная	50
		сладкий рулет	125
		борщ	175
		Итого:	350
18	12/15/2020 5:46:51 AM	борщ	175
		борщ	175
		борщ	175
		борщ	175
		борщ	175
		Итого:	875
19	12/15/2020 6:47:45 AM	сладкий рулет	125
		Итого:	125
		Всего:	1500
Расходы			
номер контракта	дата	имя продукта	название фирмы
10515	12/15/2020 2:31:47 AM	лимон	Mush
		глазурь	Classic Dungeon
		Итого:	100
11118	12/15/2020 5:36:18 AM	укроп	Prima
		Итого:	1000

Источник данных – встроенные SQL запросы для встроенных отчетов «Доходы» и «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.

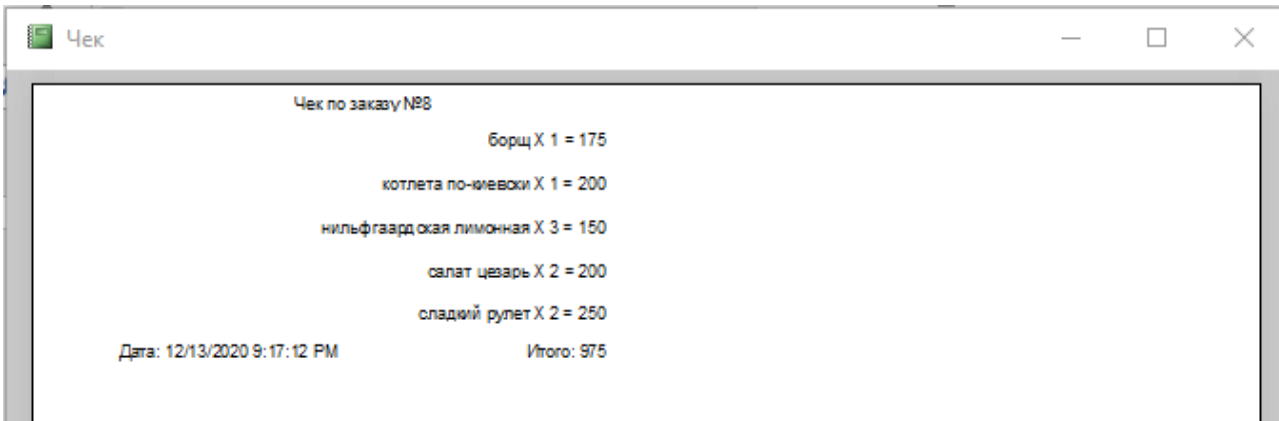
8.4. Заказ

Заказ № 8			
имя официанта	номер столика	дата	
Кузищин В. И.	8	12/13/2020 9:17:12 PM	
	количество блюда	название блюда	цена
	2	сладкий рулет	250
	2	салат цезарь	200
	3	нильгаардская лимонная	150
	1	котлета по-киевски	200
	1	борщ	175
Итого			975
Tuesday, December 15, 2020			Page 1 of 1

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];
```

8.5. Чек



Чек по заказу №8

борщ X 1 =	175
котлета по-киевски X 1 =	200
нильгаардская лимонная X 3 =	150
салат цезарь X 2 =	200
сладкий рулет X 2 =	250

Дата: 12/13/2020 9:17:12 PM Итого: 975

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.