

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Новосибирский национальный исследовательский государственный
университет»
(Новосибирский государственный университет)
Структурное подразделение Новосибирского государственного
университета –
Высший колледж информатики Университета
КАФЕДРА ИНФОРМАТИКИ

КУРСОВАЯ РАБОТА

По дисциплине: «Основы проектирования баз данных»
На тему: «Разработка базы данных для аптеки»

Студент гр. 907св1

Жаркова Е.С.

Руководитель

Иваньчева Т.А.

“ ____ ” _____ 2020 г.

Новосибирск
2020

ОГЛАВЛЕНИЕ

ПОСТАНОВКА ЗАДАЧИ	3
ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ.....	3
ГРУППЫ ПОЛЬЗОВАТЕЛЕЙ.....	3
ФУНКЦИОНАЛЬНЫЕ И НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ	4
ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ	5
СТРУКТУРА БАЗЫ ДАННЫХ.....	6
РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ.....	9
ЗАКЛЮЧЕНИЕ.....	19
ЛИСТИНГ	20
СПИСОК ИСТОЧНИКОВ	39

ПОСТАНОВКА ЗАДАЧИ

ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Требуется создать специальное программное обеспечение, которое автоматизирует работу аптеки. Для этого необходимо спроектировать базу данных и удобный интерфейс, подходящий для всех групп пользователей.

Аптека осуществляет торговлю различными препаратами.

Препараты доставляются в аптеку поставщиками. Один поставщик может поставить несколько препаратов. И один товар может поставляться разными поставщиками.

Сотрудник может найти нужный препарат, зная болезнь или симптом.

Директор может просмотреть всю информацию о сотрудниках и поставщиках и при необходимости дополнить ее. Также директор должен иметь возможность посмотреть прибыль аптеки.

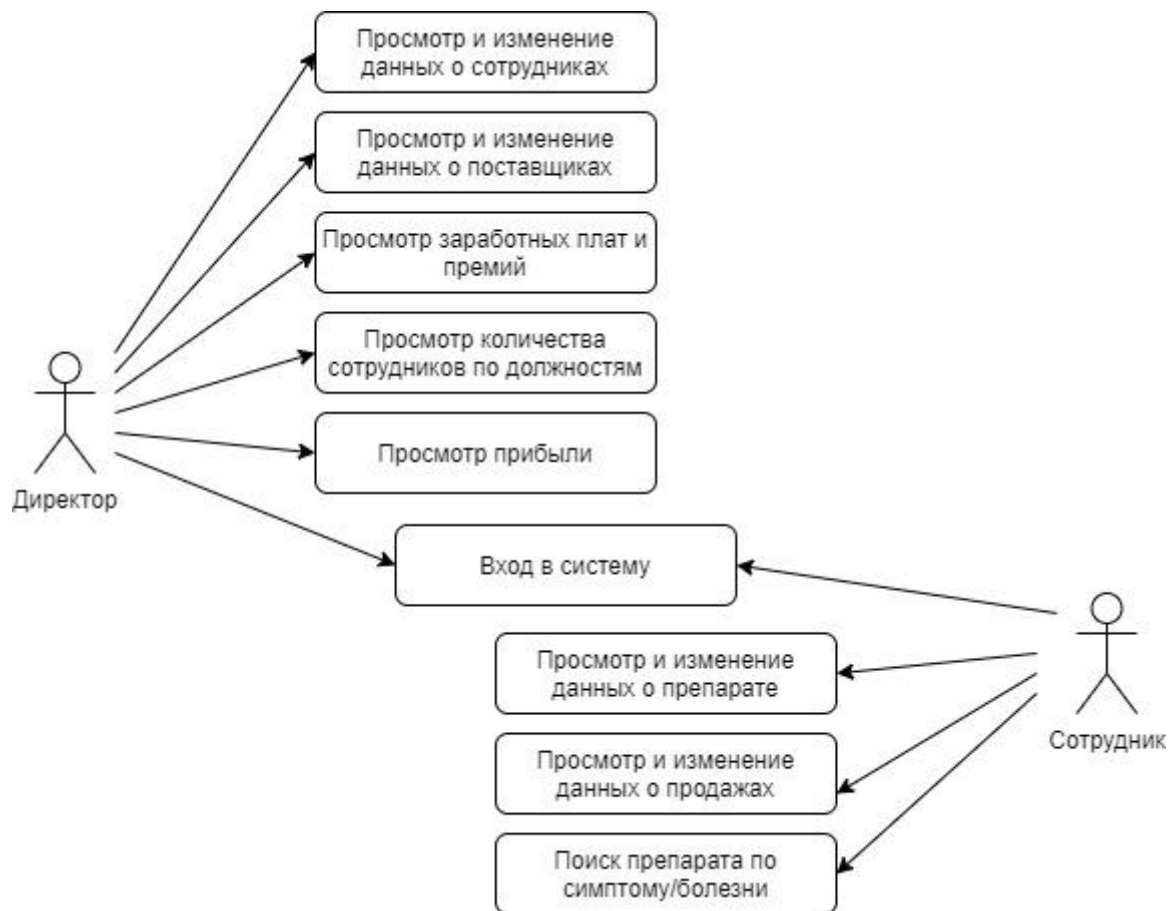
Предполагается, что данное программное обеспечение работает для одной конкретной аптеки.

ГРУППЫ ПОЛЬЗОВАТЕЛЕЙ

Работать с базой данных «Аптеки» будут следующие пользователи:

- Сотрудник аптеки
- Директор

ФУНКЦИОНАЛЬНЫЕ И НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ



Функциональные:

Сотрудник должны иметь возможность:

1. Просмотра данных всех препаратов
2. Добавления/удаления препаратов
3. По симптому/болезни узнать, какой препарат следует отпустить
4. По названию препарата узнать его противопоказания и применение
5. Узнать и добавить данные по продажам

Директор должен иметь возможность:

1. Просмотр данных о сотрудниках
2. Добавления/удаления сотрудника
3. Просмотра заработных плат и премий
4. Просмотра количества сотрудников по должностям
5. Просмотра данных о поставщиках
6. Добавления/удаления поставщиков
7. Узнать по названию препарата, кто его доставляет
8. Узнать общую прибыль и прибыль по препаратам

Нефункциональные:

1. Интерфейс (с помощью языка программирования C#)
2. Создание базы данных с помощью Microsoft SQL Server Management Studio

ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

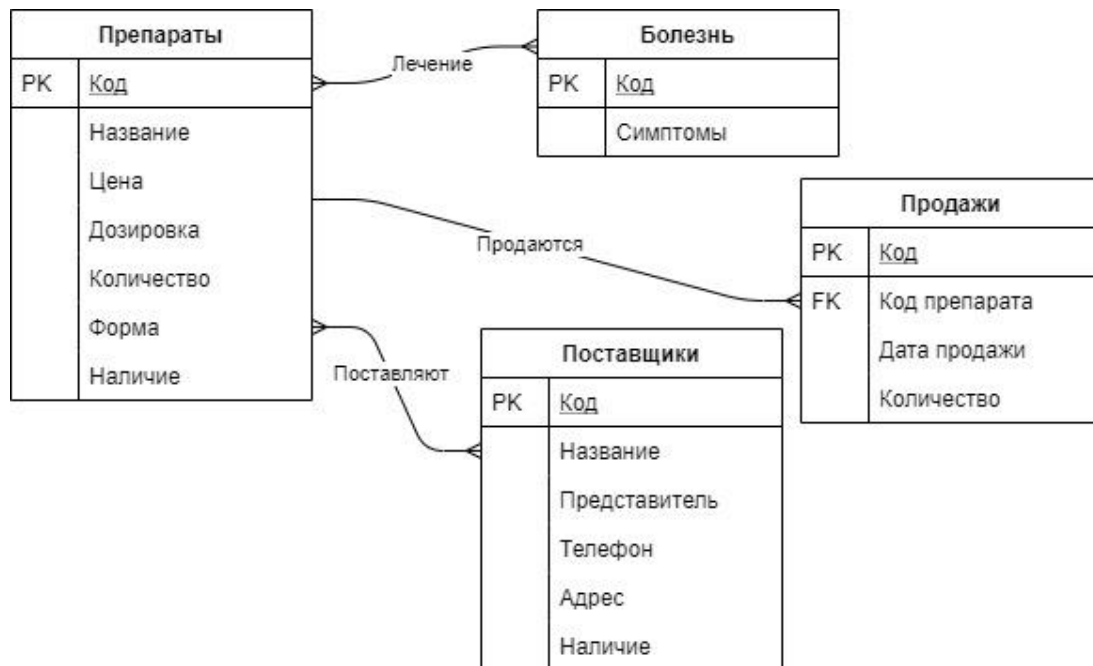
ER-модель

Сущности:

1. Препараты. Свойства сущности: Название, Цена, Дозировка, Количество, Форма, Наличие.
2. Поставщики. Свойства сущности: Название, Представитель, Телефон, Адрес.
3. Болезнь. Свойства сущности: Симптомы.
4. Продажи. Препарат, Дата продажи, Количество.

Связи:

1. Препараты - Поставщики – связь «много-к-многим»
2. Препараты – Болезнь – связь «много-к-многим»
3. Препараты – Продажи – связь «один-ко-многим»




СТРУКТУРА БАЗЫ ДАННЫХ


На основании спроектированной ER-модели получаем следующую структуру реляционной базы данных:

1. Пользователи (Код, Логин, Пароль, Роль)
2. Препараты (*Код, Название, Цена, Дозировка, Количество, Форма, Наличие)
3. Сотрудники (*Код, ФИО, ДатаРождения, Должность, Адрес, Телефон, ЗаработнаяПлата)
4. Поставщики (*Код, Название, Представитель, Телефон, Адрес)
5. Поставки (*Код, КодПоставщика, КодПрепарата, ДатаПоставки, Количество)
6. Болезнь (*Код, Симптомы)
7. Лечение (*КодБолезни, *КодПрепарата, Противопоказания, Применение)
8. Продажи(*КодПрепарата, *ДатаПродажи, Количество)


1. Пользователи

	Имя столбца	Тип данных	Разрешить ...
	Код	int	<input type="checkbox"/>
	Логин	nchar(50)	<input type="checkbox"/>
	Пароль	nvarchar(6)	<input type="checkbox"/>
	Роль	nvarchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>


2. Препараты

	Имя столбца	Тип данных	Разрешить ...
	Код	int	<input type="checkbox"/>
	Название	nvarchar(50)	<input type="checkbox"/>
	Цена	decimal(5, 2)	<input type="checkbox"/>
	Дозировка	nvarchar(50)	<input checked="" type="checkbox"/>
	Количество	nvarchar(20)	<input checked="" type="checkbox"/>
	Форма	nvarchar(20)	<input checked="" type="checkbox"/>
	Наличие	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>


3.Сотрудники

	Имя столбца	Тип данных	Разрешить ...
	Код	int	<input type="checkbox"/>
	ФИО	nvarchar(50)	<input type="checkbox"/>
	[Дата Рождения]	nvarchar(50)	<input checked="" type="checkbox"/>
	Должность	nvarchar(50)	<input type="checkbox"/>
	Телефон	nvarchar(50)	<input type="checkbox"/>
	Адрес	nvarchar(50)	<input checked="" type="checkbox"/>
	[Заработная плата]	int	<input type="checkbox"/>
			<input type="checkbox"/>


4. Поставщики

	Имя столбца	Тип данных	Разрешить ...
	Код	int	<input type="checkbox"/>
	Название	nvarchar(50)	<input type="checkbox"/>
	Представитель	nvarchar(50)	<input type="checkbox"/>
	Телефон	nvarchar(50)	<input type="checkbox"/>
	Адрес	nvarchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

5. Поставки

	Имя столбца	Тип данных	Разрешить ...
	Код	int	<input type="checkbox"/>
	[Код Поставщика]	int	<input type="checkbox"/>
	[Код Препарата]	int	<input type="checkbox"/>
	[Дата поставки]	date	<input type="checkbox"/>
	Количество	int	<input type="checkbox"/>
			<input type="checkbox"/>


6. Болезнь

	Имя столбца	Тип данных	Разрешить ...
	Код	int	<input type="checkbox"/>
	Симптомы	nvarchar(150)	<input type="checkbox"/>
			<input type="checkbox"/>

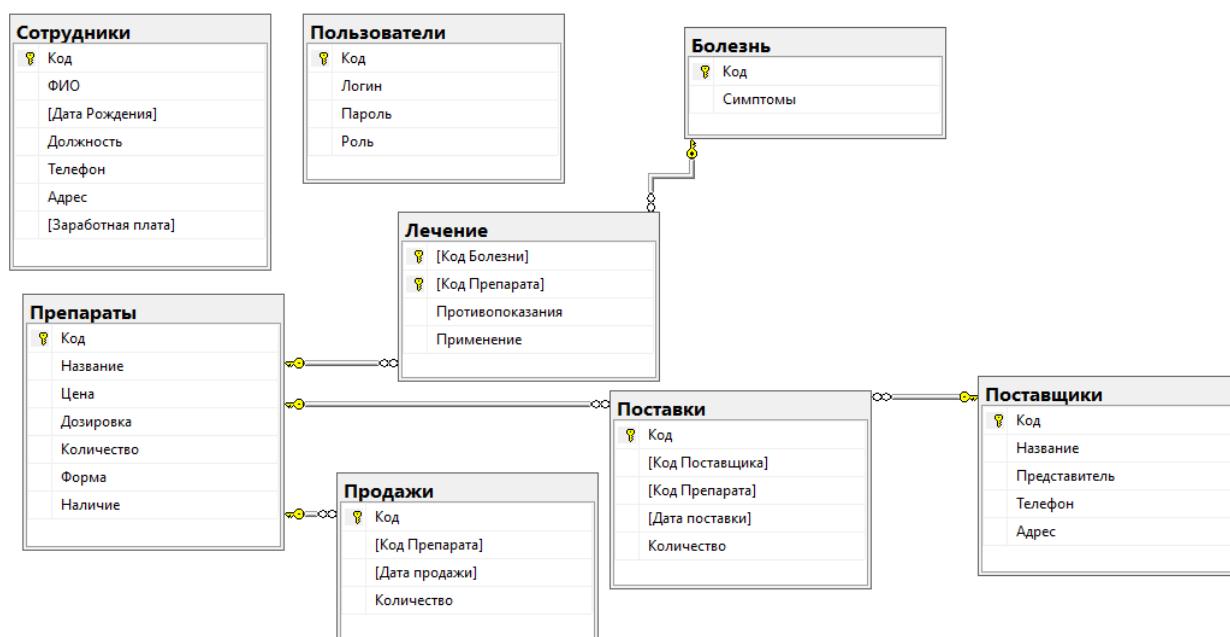
7. Лечение

	Имя столбца	Тип данных	Разрешить ...
	[Код Болезни]	int	<input type="checkbox"/>
	[Код Препарата]	int	<input type="checkbox"/>
	Противопоказания	nvarchar(100)	<input checked="" type="checkbox"/>
	Применение	nvarchar(100)	<input type="checkbox"/>
			<input type="checkbox"/>

8. Продажи

	Имя столбца	Тип данных	Разрешить ...
	Код	int	<input type="checkbox"/>
	[Код Препарата]	int	<input type="checkbox"/>
	[Дата продажи]	date	<input type="checkbox"/>
	Количество	int	<input type="checkbox"/>
			<input type="checkbox"/>

Связи между таблицами:

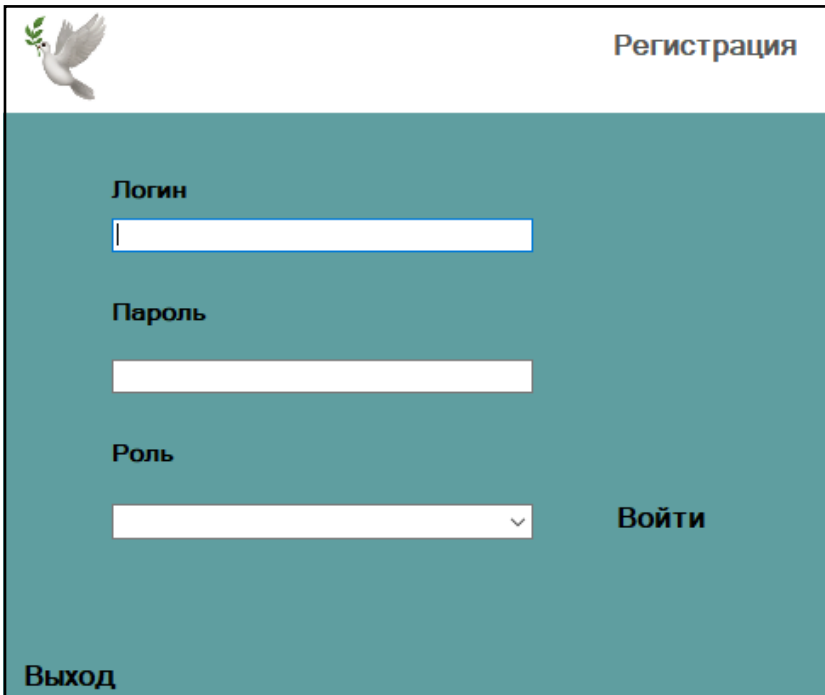


РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

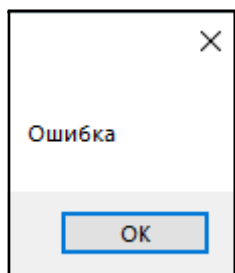
Форма входа

При запуске приложения пользователю открывается форма входа в его рабочий интерфейс.


Введите ваши данные и выберите должность. После успешной регистрации Вы сможете продолжить работу в приложении.

The screenshot shows a registration window titled "Регистрация" in the top right corner. In the top left corner, there is a logo of a white dove with an olive branch. The main area has a teal background and contains three input fields: "Логин" (Login), "Пароль" (Password), and "Роль" (Role). The "Роль" field is a dropdown menu. To the right of the "Роль" field is a "Войти" (Login) button. In the bottom left corner, there is a "Выход" (Exit) button.

В случае ошибки (неправильно введен пароль/логин/должность) на экран будет выведено соответствующее сообщение.



Если Вы новый сотрудник, выберете на форме должность “Сотрудник”, после чего Вам станет доступно окно регистрации.



Регистрация

Логин

Пароль


Роль

Сотрудник

Войти

Выход

Введите ваши данные и нажмите кнопку «Зарегистрироваться», после чего Вы будете перенаправлены вновь на окно входа, где Вы уже можете ввести свои данные и войти.



Регистрация

Логин

Пароль

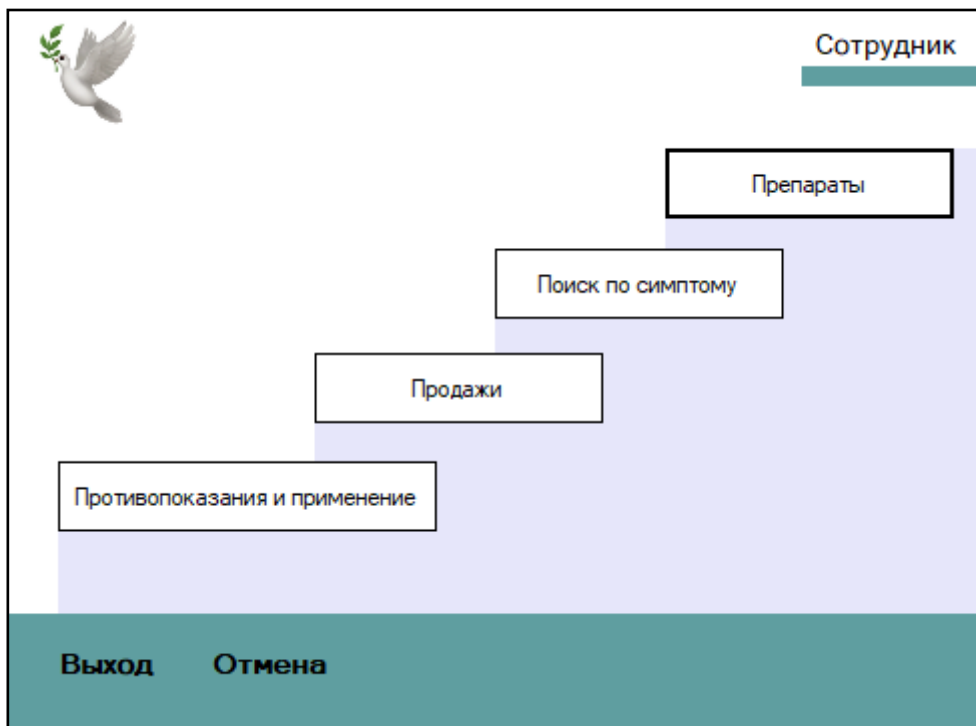
Зарегистрировать

Выход

Отмена

Форма «Сотрудник»

Если Вы являетесь сотрудником, Вы будете перенаправлены на соответствующее окно. Здесь Вы можете выбрать тот сегмент, который Вам необходим. Для этого нажмите на необходимую кнопку.



В окне «Препараты» Вы можете добавлять/удалять препарат. Или же изменять уже существующие данные. Но не забывайте сохранять введенные данные.

Все это Вы можете сделать с помощью навигационной панели.

1 из 14

Препараты

	Название	Цена	Дозировка	Количество	Форма	Наличие
▶	Парацетамол Экспресс	93,00	0,5+0,15	10 шт.	таблетки	34
	Парацетамол м...	25,00	0,5	10 шт.	таблетки	45
	Антигриппин	350,00	4,36	10 шт.	порошок	34
	Антигрипп ком...	400,00	5	10 шт.	порошок	23
	Анвимакс	270,00	3	20 шт.	порошок	45
	Анвимакс ультра	190,00	5	10 шт.	порошок	13
	Супрадин	360,00		10 шт.	таблетки	23
	Супрастин	127,00	0,025	20 шт.	таблетки	54
	Пиковит	380,00		30 шт.	таблетки	52
	Компливит	178,00		30 шт.	таблетки	29
	Стодаль	315,00	200		сироп	19

Название: Парацетамол Экспресс

Цена: 93,00

Дозировка: 0,5+0,15

Количество: 10 шт.

Форма: таблетки

Наличие: 34

Выход Отмена

В окне «Поиск препарата по симптому» Вы можете найти нужный препарат, указав симптом (регистр не учитывается) и нажав кнопку «Найти».

Поиск препарата по симптому

Симптом

	Симптомы	Название	Дозировка	Форма	Цена	Количество
▶	жар, воспаление	Парацетамол Экст...	0,5+0,15	таблетки	93,00	10 шт.
	жар, воспаление	Парацетамол мед...	0,5	таблетки	25,00	10 шт.
	жар, воспаление	Кетонал	0,05+2	капсулы	235,00	10 шт.
	ОРВИ, грипп, жар,...	Антигриппин	4,36	порошок	350,00	10 шт.
	ОРВИ, грипп, жар,...	Антигрипп комплекс	5	порошок	400,00	10 шт.
	ОРВИ, грипп, жар,...	Анвимакс	3	порошок	270,00	20 шт.
	ОРВИ, грипп, жар,...	Анвимакс ультра	5	порошок	190,00	10 шт.
*						

В окне «Продажи» Вы можете посмотреть данные о проданных препаратах, а также добавить/удалить/изменить их с помощью навигационной панели.

Препарат указывается в виде его идентификатора (кода препарата). Узнать код препарата по названию или же, наоборот, название по коду можно с помощью представленной функции поиска. Для этого в соответствующее поле введите нужный параметр и нажмите на кнопку нужного поиска.

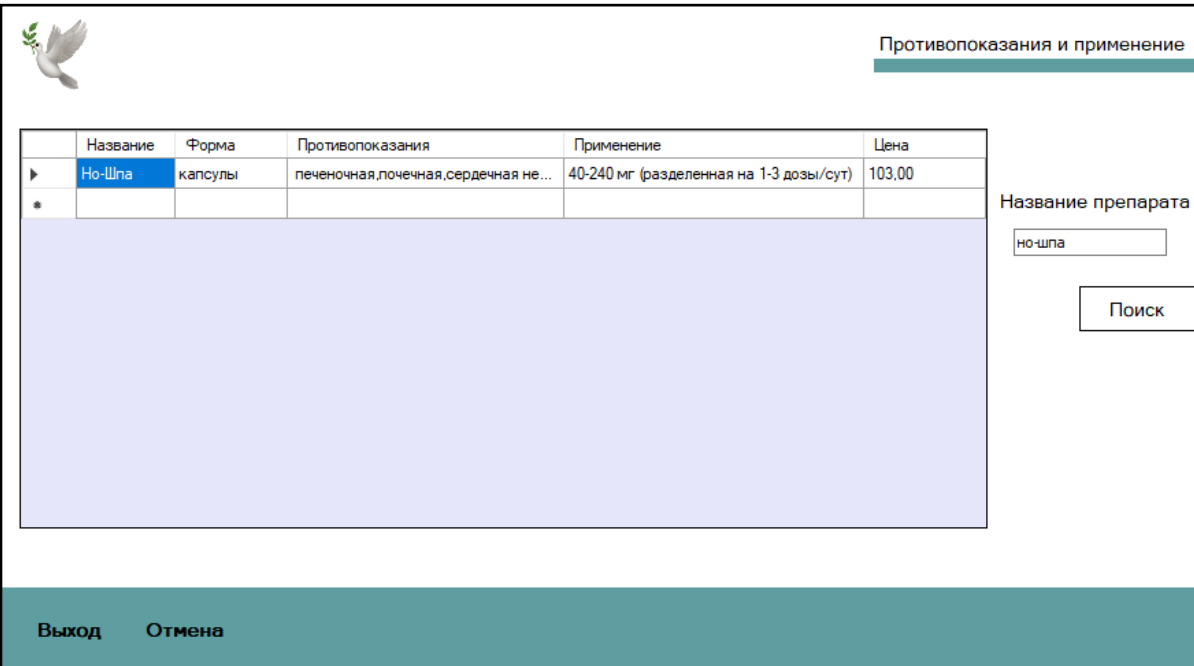
Продажи

	Код Препарата	Дата продажи	Количество
▶	3	01.05.2020	3
	4	03.05.2020	1
	5	01.05.2020	1
	6	05.05.2020	2
	8	01.05.2020	3
	9	06.05.2020	5
	10	04.05.2020	3
	12	08.05.2020	4

Код Препарата:
Дата продажи:
Количество:

	Название	Код
▶	Бронхикум	16
*		

В окне «Противопоказания и применение» Вы можете узнать эти параметры у конкретного препарата с помощью поиска (укажите нужное название и нажмите на кнопку «поиск»)



Противопоказания и применение

	Название	Форма	Противопоказания	Применение	Цена
▶	Но-Шпа	капсулы	печеночная, почечная, сердечная не...	40-240 мг (разделенная на 1-3 дозы/сут)	103,00
*					

Название препарата


Форма «Директор»

Если Вы являетесь директором, Вы будете перенаправлены на соответствующее окно.

Если Вам необходимо узнать о прибыли организации, следует нажать на кнопку «Общие суммы продаж» (уже при наведении на поле Вы увидите, что кнопка выделится цветом).

Если Вам необходимо узнать/изменить информацию о сотрудниках или поставщиках, выберите соответствующий элемент (напротив выбранного элемента кружок маркируется черным цветом) и нажмите кнопку «Продолжить».

После чего Вам откроется соответствующее окно.



Директор

Выберите, о ком вы хотите получить информацию


☒ Сотрудники
☐ Поставщики

Продолжить

Общие суммы продаж

Выход Отмена

В окне «Общие суммы продаж» вы можете узнать нужную информацию, используя кнопки «Заполнить».



Сумма продаж по препаратам

	Общая сумма продаж	Название	Количество
▶	350,00	Антигриппин	1
	390,00	Бронхикум	1
	25,00	Парацетамол м...	1
	800,00	Антигрипп комп...	2
	760,00	Пиковит	2
	1260,00	Стодаль	2

Заполнить

Общий доход

	Общий доход
▶	10748,00
*	

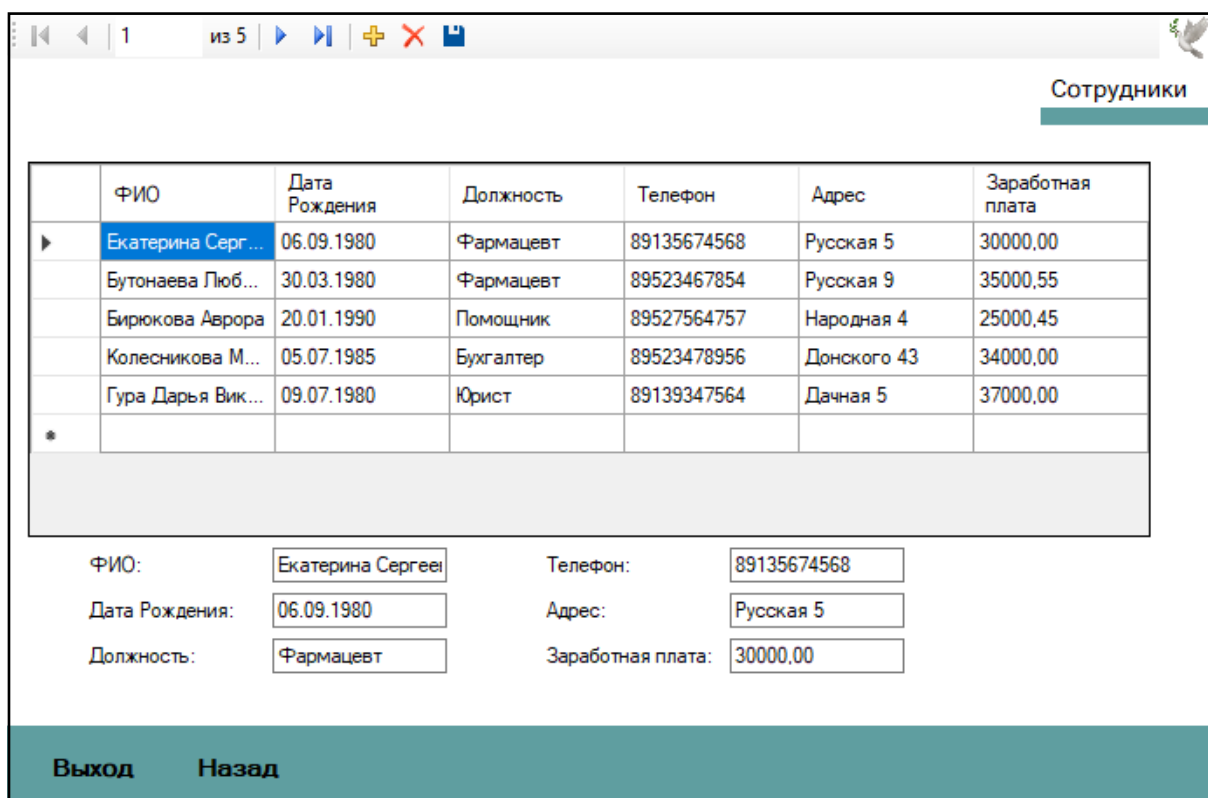
Заполнить

Выход Назад

В окне «Работа с сотрудниками» Вам предложено выбрать кнопку, соответствующую необходимой Вам информации.



В окне «Сотрудники» Вы можете узнать полную информацию о своих сотрудниках. А также изменить или добавить/удалить эту информацию с помощью навигационной панели.



В окне «Заработные платы и премии» Вы можете увидеть доходы всех сотрудников.

Для этого необходимо нажать кнопку «Заполнить», после чего вся информация будет отображена в таблице.



Заработные платы и премии

	ФИО	Должность	Заработная плата	Премия	Итоговая з.п.
▶	Екатерина Сергее...	Фармацевт	30000,00	7500,0000	37500,0000
	Бутонаева Любовь	Фармацевт	35000,55	8750,1375	43750,6875
	Бирюкова Аврора	Помощник	25000,45	6250,1125	31250,5625
	Колесникова Мар...	Бухгалтер	34000,00	8500,0000	42500,0000
	Гура Дарья Викто...	Юрист	37000,00	9250,0000	46250,0000
*					

Заполнить

Выход

Отмена

В окне «Личные данные» представлены данные сотрудников, такие как дата рождения, адрес и телефон. Отобразить данные в таблице можно нажатием кнопки «Заполнить».



Личные данные


	ФИО	Дата рождения	Адрес	Телефон
▶	Екатерина Серг...	06.09.1980	Русская 5	89135674568
	Бутонаева Любо...	30.03.1980	Русская 9	89523467854
	Бирюкова Аврора	20.01.1990	Народная 4	89527564757
	Колесникова М...	05.07.1985	Донского 43	89523478956
	Гура Дарья Вик...	09.07.1980	Дачная 5	89139347564
*				

Заполнить

Выход

Отмена

В окне «Количество сотрудников» Вы можете узнать, сколько сотрудников находятся на каждой должности. Заполните таблицу нажатием кнопки «Заполнить».




Количество сотрудников

	Должность	Количество сотрудников
▶	Бухгалтер	1
	Помощник	1
	Фармацевт	2
	Юрист	1
*		

Заполнить

Выход Назад

Если на главной форме Директора Вы выбрали «работу с поставщиками», Вам откроется форма, где Вы можете продолжить работу в соответствии с необходимой Вам информацией. Нажмите на необходимую Вам кнопку.



Работа с поставщиками

Поставщики

Поиск поставщиков

Выход Отмена

В окне «Поставщики» Вы можете управлять информацией о поставщиках с помощью навигационной панели.

Поставщики

	Название	Представитель	Телефон	Адрес
▶	Хиноин-Завод	Валерьянов Сер...	567854	Ленина 6
	Фармстандарт	Здоровенко Ни...	784567	Кисловодская 8
	Синтез	Неболейка Ана...	983456	Алчхита 7
	Жить Здорово	Малышева Елен...	233244	Эрнста 1
*				

Название:

Представитель:

Телефон:

Адрес:

Выход **Отмена**

В окне «Поиск поставщиков» Вы можете по названию препарата узнать, какие поставщики его поставляют. Для этого необходимо ввести в поле название препарата и нажать кнопку найти.

Поиск поставщиков по препарату

Название препарата

	Название	Форма	Поставщик	Адрес	Телефон	Дата поставки	Количество
▶	Стодаль	сироп	Хиноин-Завод	Ленина 6	567854	03.02.2020	34
*							

Выход **Отмена**

Кнопки «Выход» и «Отмена» находятся на всех окнах и работают одинаково.

«Выход» подразумевает полный выход из приложения.

«Отмена» возвращает Вас на предыдущее окно.

ЗАКЛЮЧЕНИЕ

Также при необходимости могут быть внесены дополнительные функциональные возможности пользователей. Например,

1. Возможность поиска препаратов по нескольким параметрам.
2. Возможность вывода самых продаваемых товаров.
3. Возможность изменения процента, по которому начисляется премия.

ЛИСТИНГ

- CREATE TABLE [dbo].[**Пользователи**]
([Код] [int] IDENTITY(1,1) NOT NULL,
[Логин] [nvarchar](50) NOT NULL,
[Пароль] [nvarchar](6) NOT NULL,
[Роль] [nvarchar](50) NOT NULL,
CONSTRAINT [PK_Пользователи] PRIMARY KEY CLUSTERED
([Код] ASC))
- CREATE TABLE [dbo].[**Препараты**]
([Код] [int] IDENTITY(1,1) NOT NULL,
[Название] [nvarchar](50) NOT NULL,
[Цена] [decimal](5, 2) NOT NULL,
[Дозировка] [nvarchar](50) NULL,
[Количество] [nvarchar](20) NULL,
[Форма] [nvarchar](20) NULL,
[Наличие] [int] NULL,
CONSTRAINT [PK_Препараты] PRIMARY KEY CLUSTERED
([Код] ASC))
- CREATE TABLE [dbo].[**Сотрудники**]
([Код] [int] IDENTITY(1,1) NOT NULL,
[ФИО] [nvarchar](50) NOT NULL,
[Дата Рождения] [nvarchar](50) NULL,
[Должность] [nvarchar](50) NOT NULL,
[Телефон] [nvarchar](50) NOT NULL,
[Адрес] [nvarchar](50) NULL,
[Заработная плата] [int] NOT NULL,
CONSTRAINT [PK_Сотрудники] PRIMARY KEY CLUSTERED
([Код] ASC))
- CREATE TABLE [dbo].[**Поставщики**]
([Код] [int] IDENTITY(1,1) NOT NULL,
[Название] [nvarchar](50) NOT NULL,
[Представитель] [nvarchar](50) NOT NULL,
[Телефон] [nvarchar](50) NOT NULL,
[Адрес] [nvarchar](50) NULL,
CONSTRAINT [PK_Поставщики] PRIMARY KEY CLUSTERED
([Код] ASC))
- CREATE TABLE [dbo].[**Поставки**]
([Код] [int] IDENTITY(1,1) NOT NULL,
[Код Поставщика] [int] NOT NULL,
[Код Препарата] [int] NOT NULL,
[Дата поставки] [date] NOT NULL,
[Количество] [int] NOT NULL,
CONSTRAINT [PK_Поставки] PRIMARY KEY CLUSTERED
([Код] ASC))
ALTER TABLE [dbo].[Поставки] WITH CHECK ADD CONSTRAINT
[FK_Поставки_Поставщики] FOREIGN KEY([Код Поставщика])

```
REFERENCES [dbo].[Поставщики] ([Код])
GO
ALTER TABLE [dbo].[Поставки] WITH CHECK ADD CONSTRAINT
[FK_Поставки_Препараты] FOREIGN KEY([Код Препарата])
REFERENCES [dbo].[Препараты] ([Код])
GO
```

- CREATE TABLE [dbo].[**Болезнь**]
([Код] [int] IDENTITY(1,1) NOT NULL,
[Симптомы] [nvarchar](150) NOT NULL,
CONSTRAINT [PK_Болезнь] PRIMARY KEY CLUSTERED
([Код] ASC))

- CREATE TABLE [dbo].[**Лечение**]
([Код Болезни] [int] NOT NULL,
[Код Препарата] [int] NOT NULL,
[Противопоказания] [nvarchar](100) NULL,
[Применение] [nvarchar](100) NULL,
CONSTRAINT [PK_Лечение] PRIMARY KEY CLUSTERED
([Код Болезни] ASC, [Код Препарата] ASC))

```
ALTER TABLE [dbo].[Лечение] WITH CHECK ADD CONSTRAINT
[FK_Лечение_Болезнь] FOREIGN KEY([Код Болезни])
REFERENCES [dbo].[Болезнь] ([Код])
GO
ALTER TABLE [dbo].[Лечение] WITH CHECK ADD CONSTRAINT
[FK_Лечение_Препараты] FOREIGN KEY([Код Препарата])
REFERENCES [dbo].[Препараты] ([Код])
GO
```

- CREATE TABLE [dbo].[**Продажи**]
([Код] [int] IDENTITY(1,1) NOT NULL,
[Код Препарата] [int] NOT NULL,
[Дата продажи] [date] NOT NULL,
[Количество] [int] NOT NULL,
CONSTRAINT [PK_Продажи] PRIMARY KEY CLUSTERED
([Код] ASC))

```
ALTER TABLE [dbo].[Продажи] WITH CHECK ADD CONSTRAINT
[FK_Продажи_Препараты] FOREIGN KEY([Код Препарата])
REFERENCES [dbo].[Препараты] ([Код])
GO
```

Форма входа/Решение FormEnter.cs

```
namespace АРТЕКА
{
    public partial class FormEnter : Form
    {
```

```

public static string strConn = @"Data Source=DESKTOP-
Q5N5GV\SQLEXPRESS;Initial Catalog=аптека;Integrated Security=True";

public FormEnter()
{
    InitializeComponent();
}

private void Exit_Click(object sender, EventArgs e)
{
    Application.Exit();
}

private void ComboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    if (ComboBox1.Items[ComboBox1.SelectedIndex].ToString() == "Сотрудник")
    {
        Reg.Enabled = true;
    }
    else Reg.Enabled = false;
}

private void Enter_Click(object sender, EventArgs e)
{
    if ((texlogin.Text == "") || (texpass.Text == ""))
    {
        MessageBox.Show("Введите данные");
        return;
    }
    if (ComboBox1.SelectedIndex == -1)
    {
        MessageBox.Show("Выберите роль");
        return;
    }
    string role = ComboBox1.Items[ComboBox1.SelectedIndex].ToString();
    string login = texlogin.Text;
    string password = texpass.Text;

    string sql = "SELECT * FROM [Пользователи] WHERE [Логин]='"+login+"' and
[Пароль]='"+password+"' and [Роль]='"+role+"'";

    using (SqlConnection conn = new SqlConnection(strConn))
    {
        conn.Open();
        SqlCommand comm = new SqlCommand(sql, conn);
        SqlDataReader rd = comm.ExecuteReader();
        if (rd.HasRows)
        {
            MessageBox.Show("Вход успешно выполнен");
            switch (role)

```

```

        {
            case "Директор":
                FormDirector director = new FormDirector();
                this.Hide();
                director.Show();
                break;
            case "Сотрудник":
                FormEmployee employee = new FormEmployee();
                this.Hide();
                employee.Show();
                break;
        }
    }
    else
    {
        MessageBox.Show("Ошибка");
    }
}

private void Reg_Click(object sender, EventArgs e)
{
    FormReg rg = new FormReg();
    this.Hide();
    rg.Show();
}
}
}

```

Форма Регистрации/Решение FormReg.cs

```

namespace АРТЕКА
{
    public partial class FormReg : Form
    {
        string strConn = FormEnter.strConn;

        public FormReg()
        {
            InitializeComponent();
        }

        private void Cancel_Click(object sender, EventArgs e)
        {
            this.Close();
            FormEnter mn = new FormEnter();
            mn.Show();
        }

        private void Exit_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }
    }
}

```

```

    }

    private void Register_Click(object sender, EventArgs e)
    {
        if ((texreglog.Text == "") || (texregpass.Text == ""))
        {
            MessageBox.Show("Введите данные");
            return;
        }
        else
        {
            string login = texreglog.Text;
            string password = texregpass.Text;
            string sql = "INSERT INTO [Пользователи] values ('"+login+"', '"+password+"',
'Sотрудник') ";

            using (SqlConnection conn = new SqlConnection(strConn))
            {
                conn.Open();
                SqlCommand comm = new SqlCommand(sql, conn);
                try
                {
                    comm.ExecuteNonQuery();
                }
                catch (Exception ex)
                {
                    MessageBox.Show(ex.Message);
                    return;
                }
                MessageBox.Show("Вы успешно зарегистрированы!");
                FormEnter main = new FormEnter();
                this.Hide();
                main.Show();
            }
        }
    }
}

```

Форма Сотрудника/Решение FormEmployee.cs

```

namespace АРТЕКА
{
    public partial class FormEmployee : Form
    {
        public FormEmployee()
        {
            InitializeComponent();
        }

        private void Exit_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }
    }
}

```



```

    }

    private void Preparat_Click(object sender, EventArgs e)
    {
        Medicines pr = new Medicines();
        this.Hide();
        pr.Show();
    }

    private void Cancel_Click(object sender, EventArgs e)
    {
        this.Close();
        FormEnter mn = new FormEnter();
        mn.Show();
    }

    private void Sales_Click(object sender, EventArgs e)
    {
        Sales sl = new Sales();
        this.Hide();
        sl.Show();
    }

    private void Search_Click_1(object sender, EventArgs e)
    {
        SearchSymptom sr = new SearchSymptom();
        this.Hide();
        sr.Show();
    }

    private void Protivop_Click(object sender, EventArgs e)
    {
        SearchProtiv ab = new SearchProtiv();
        this.Hide();
        ab.Show();
    }
}
}

```

Форма Препаратов/Решение Medicines.cs

```

namespace APTEKA
{
    public partial class Medicines : Form
    {
        string strConn = FormEnter.strConn;

        public Medicines()
        {
            InitializeComponent();
        }
    }
}

```

```

private void FormEmployee1_Load(object sender, EventArgs e)
{
    // данная строка кода позволяет загрузить данные в таблицу
    "аптекаDataSet3.Препараты".
    this.препаратыTableAdapter.Fill(this.аптекаDataSet3.Препараты);
}

private void Exit_Click(object sender, EventArgs e)
{
    Application.Exit();
}

private void Cancel_Click_1(object sender, EventArgs e)
{
    this.Close();
    FormEmployee em = new FormEmployee();
    em.Show();
}

private void ПрепаратыBindingNavigatorSaveItem_Click(object sender, EventArgs e)
{
    this.Validate();
    this.препаратыBindingSource.EndEdit();
    this.tableAdapterManager.UpdateAll(this.аптекаDataSet3);
}
}
}

```

Форма поиска по симптому/Решение SearchSymptom.cs

```

namespace АПТЕКА
{
    public partial class SearchSymptom : Form
    {
        string strConn = FormEnter.strConn;
        SqlDataAdapter ada;
        DataSet ds;

        public SearchSymptom()
        {
            InitializeComponent();
        }

        private void Exit_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void Cancel_Click(object sender, EventArgs e)
        {

```

```

        this.Close();
        FormEmployee em = new FormEmployee();
        em.Show();
    }

    private void Button1_Click(object sender, EventArgs e)
    {
        using (SqlConnection conn = new SqlConnection(strConn))
        {
            conn.Open();
            SqlCommand comm = new SqlCommand("SELECT Болезнь.Симптомы,
Препараты.Название, Препараты.Дозировка, Препараты.Форма, Препараты.Цена,
Препараты.Количество FROM Препараты INNER JOIN(Болезнь INNER JOIN Лечение
ON Болезнь.Код = Лечение.[Код Болезни]) ON Препараты.Код = Лечение.[Код
Препарата] WHERE Болезнь.Симптомы LIKE '%" + textBox1.Text.ToString() + "%'",
conn);
            ada = new SqlDataAdapter(comm);
            ds = new DataSet();
            ada.Fill(ds);
            dataGridView1.DataSource = ds.Tables[0];
        }
    }
}

```

Форма продаж/Решение Sales.cs

```

namespace АПТЕКА
{
    public partial class Sales : Form
    {
        string strConn = FormEnter.strConn;
        SqlDataAdapter ada;
        DataSet ds;
        public Sales()
        {
            InitializeComponent();
        }

        private void ПродажиBindingNavigatorSaveItem_Click(object sender, EventArgs e)
        {
            this.Validate();
            this.продажиBindingSource.EndEdit();
            this.tableAdapterManager.UpdateAll(this.аптекаDataSet2);
        }

        private void Sales_Load(object sender, EventArgs e)
        {
            // данная строка кода позволяет загрузить данные в таблицу
            "аптекаDataSet2.Продажи".
            this.продажиTableAdapter.Fill(this.аптекаDataSet2.Продажи);
        }
    }
}

```

```

{
    public partial class SearchProtiv : Form
    {
        string strConn = FormEnter.strConn;
        SqlDataAdapter ada;
        DataSet ds;

        public SearchProtiv()
        {
            InitializeComponent();
        }

        private void Exit_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void Cancel_Click(object sender, EventArgs e)
        {
            this.Close();
            FormEmployee em = new FormEmployee();
            em.Show();
        }

        private void Button1_Click(object sender, EventArgs e)
        {
            using (SqlConnection conn = new SqlConnection(strConn))
            {
                conn.Open();
                SqlCommand comm = new SqlCommand("SELECT Препараты.Название,
Препараты.Форма, Лечение.Противопоказания, Лечение.Применение, Препараты.Цена
FROM Препараты INNER JOIN Лечение ON Препараты.Код = Лечение.[Код
Препарата] WHERE Препараты.Название LIKE '%" + textBox1.Text.ToString() + "%'",
conn);
                ada = new SqlDataAdapter(comm);
                ds = new DataSet();
                ada.Fill(ds);
                dataGridView1.DataSource = ds.Tables[0];
            }
        }
    }
}

```

Форма Директора/Решение FormDirector.cs

```

namespace АПТЕКА
{
    public partial class FormDirector : Form
    {

```

```

    }

    private void Button1_Click(object sender, EventArgs e)
    {
        using (SqlConnection conn = new SqlConnection(strConn))
        {
            conn.Open();
            SqlCommand comm = new SqlCommand("SELECT Название, Код FROM
Препараты WHERE Препараты.Код LIKE '%" + textBox1.Text.ToString() + "%'", conn);
            ada = new SqlDataAdapter(comm);
            ds = new DataSet();
            ada.Fill(ds);
            dataGridView1.DataSource = ds.Tables[0];
        }
    }

    private void Button2_Click(object sender, EventArgs e)
    {
        using (SqlConnection conn = new SqlConnection(strConn))
        {
            conn.Open();
            SqlCommand comm = new SqlCommand("SELECT Название, Код FROM
Препараты WHERE Препараты.Название LIKE '%" + textBox2.Text.ToString() + "%'",
conn);
            ada = new SqlDataAdapter(comm);
            ds = new DataSet();
            ada.Fill(ds);
            dataGridView1.DataSource = ds.Tables[0];
        }
    }

    private void Button3_Click(object sender, EventArgs e)
    {
        this.Close();
        FormEmployee mn = new FormEmployee();
        mn.Show();
    }

    private void Button4_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }
}
}

```

Форма противопоказаний и применений/Решение SearchProtiv.cs

namespace АПТЕКА

```

public FormDirector()
{
    InitializeComponent();
}

private void Exit_Click(object sender, EventArgs e)
{
    Application.Exit();
}

private void Button1_Click(object sender, EventArgs e)
{
    if (radioButton1.Checked)
    {
        FormDEmployee emp = new FormDEmployee();
        this.Hide();
        emp.Show();
    }
    if (radioButton2.Checked)
    {
        FormProvider pr = new FormProvider();
        this.Hide();
        pr.Show();
    }
}

private void Cancel_Click(object sender, EventArgs e)
{
    this.Close();
    FormEnter mn = new FormEnter();
    mn.Show();
}

private void Sum_Click(object sender, EventArgs e)
{
    Sum sm = new Sum();
    this.Hide();
    sm.Show();
}
}
}

```

Форма Суммы продаж/Решение Sum.cs

```

namespace APTEKA
{
    public partial class Sum : Form
    {
        string strConn = FormEnter.strConn;
        SqlDataAdapter ada;
        DataSet ds;
        public Sum()

```

```

{
    InitializeComponent();
}

private void Button2_Click(object sender, EventArgs e)
{
    using (SqlConnection conn = new SqlConnection(strConn))
    {
        conn.Open();
        SqlCommand comm = new SqlCommand("SELECT
Sum(Препараты.Цена*Продажи.Количество) AS 'Общий доход' FROM Препараты
INNER JOIN Продажи ON Препараты.Код = Продажи.[Код препарата]", conn);
        ada = new SqlDataAdapter(comm);
        ds = new DataSet();
        ada.Fill(ds);
        dataGridView2.DataSource = ds.Tables[0];
    }
}

private void Button1_Click(object sender, EventArgs e)
{
    using (SqlConnection conn = new SqlConnection(strConn))
    {
        conn.Open();
        SqlCommand comm = new SqlCommand("SELECT
Sum(Препараты.Цена*Продажи.Количество) AS 'Общая сумма продаж',
Препараты.Название, Продажи.Количество FROM Препараты INNER JOIN Продажи
ON Препараты.Код = Продажи.[Код препарата] GROUP BY Препараты.Название,
Продажи.Количество ", conn);
        ada = new SqlDataAdapter(comm);
        ds = new DataSet();
        ada.Fill(ds);
        dataGridView1.DataSource = ds.Tables[0];
    }
}

private void Button3_Click(object sender, EventArgs e)
{
    Application.Exit();
}

private void Button4_Click(object sender, EventArgs e)
{
    this.Close();
    FormDirector dir = new FormDirector();
    dir.Show();
}
}
}

```

Форма Работа с сотрудниками/Решение FormDEmployee

```

namespace APTEKA
{
    public partial class FormDEmployee : Form
    {

        public FormDEmployee()
        {
            InitializeComponent();
        }

        private void Employee_Click(object sender, EventArgs e)
        {
            DirEmployee em = new DirEmployee();
            this.Hide();
            em.Show();
        }

        private void Salary_Click(object sender, EventArgs e)
        {
            Salary sl = new Salary();
            this.Hide();
            sl.Show();
        }

        private void Numberof_Click(object sender, EventArgs e)
        {
            AmountEmployee nm = new AmountEmployee();
            this.Hide();
            nm.Show();
        }

        private void Personal_Click(object sender, EventArgs e)
        {
            PersonalData pl = new PersonalData();
            this.Hide();
            pl.Show();
        }

        private void Exit_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void Cancel_Click(object sender, EventArgs e)
        {
            this.Close();
            FormDirector dr = new FormDirector();
            dr.Show();
        }
    }
}

```


Форма Сотрудников (для Директора)/Решение DirEmployee.cs

```
namespace АРТЕКА
{
    public partial class DirEmployee : Form
    {
        public DirEmployee()
        {
            InitializeComponent();
        }

        private void СотрудникиBindingNavigatorSaveItem_Click(object sender, EventArgs e)
        {
            this.Validate();
            this.сотрудникиBindingSource.EndEdit();
            this.tableAdapterManager.UpdateAll(this.аптекаDataSet1);
        }

        private void DirEmployee_Load(object sender, EventArgs e)
        {
            // данная строка кода позволяет загрузить данные в таблицу
            "аптекаDataSet1.Сотрудники".
            this.сотрудникиTableAdapter.Fill(this.аптекаDataSet1.Сотрудники);
        }

        private void Cancel_Click(object sender, EventArgs e)
        {
            this.Close();
            FormDEmployee de = new FormDEmployee();
            de.Show();
        }

        private void Exit_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }
    }
}
```

Форма заработных плат/Решение Salary.cs

```
namespace АРТЕКА
{
    public partial class Salary : Form
    {
        string strConn = FormEnter.strConn;
        SqlDataAdapter ada;
        DataSet ds;
        public Salary()
        {
            InitializeComponent();
        }

        private void Exit_Click(object sender, EventArgs e)
```

```

    {
        Application.Exit();
    }

    private void Cancel_Click(object sender, EventArgs e)
    {
        this.Close();
        FormDEmployee de = new FormDEmployee();
        de.Show();
    }

    private void Button1_Click(object sender, EventArgs e)
    {
        using (SqlConnection conn = new SqlConnection(strConn))
        {
            conn.Open();
            SqlCommand comm = new SqlCommand("SELECT ФИО, Должность,
[Заработная плата], [Заработная плата]*0.25 as 'Премия', [Заработная
плата]+[Заработная плата]*0.25 AS 'Итоговая з.п.' FROM Сотрудники", conn);
            ada = new SqlDataAdapter(comm);
            ds = new DataSet();
            ada.Fill(ds);
            dataGridView1.DataSource = ds.Tables[0];
        }
    }
}

```

Форма личных данных сотрудников/Решение PersonalData.cs

```

namespace АПТЕКА
{
    public partial class PersonalData : Form
    {
        string strConn = FormEnter.strConn;
        SqlDataAdapter ada;
        DataSet ds;
        public PersonalData()
        {
            InitializeComponent();
        }

        private void Exit_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void Cancel_Click(object sender, EventArgs e)
        {
            this.Close();
            FormDEmployee de = new FormDEmployee();
            de.Show();
        }
    }
}

```

```

    }

    private void Button1_Click(object sender, EventArgs e)
    {
        using (SqlConnection conn = new SqlConnection(strConn))
        {
            conn.Open();
            SqlCommand comm = new SqlCommand("SELECT ФИО,[Дата рождения],
Адрес,Телефон FROM Сотрудники", conn);
            ada = new SqlDataAdapter(comm);
            ds = new DataSet();
            ada.Fill(ds);
            dataGridView1.DataSource = ds.Tables[0];
        }
    }
}
}
}

```

Форма подсчета количества сотрудников/Решение AmountEmployee.cs

```

namespace АПТЕКА
{
    public partial class AmountEmployee : Form
    {
        string strConn = FormEnter.strConn;
        SqlDataAdapter ada;
        DataSet ds;
        public AmountEmployee()
        {
            InitializeComponent();
        }

        private void Exit_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void Cancel_Click(object sender, EventArgs e)
        {
            this.Close();
            FormDEmployee de = new FormDEmployee();
            de.Show();
        }

        private void Button1_Click(object sender, EventArgs e)
        {
            using (SqlConnection conn = new SqlConnection(strConn))
            {
                conn.Open();
                SqlCommand comm = new SqlCommand("SELECT Сотрудники.Должность,
Count(Сотрудники.ФИО) AS [Количество сотрудников] FROM Сотрудники GROUP BY
Сотрудники.Должность", conn);
                ada = new SqlDataAdapter(comm);
                ds = new DataSet();
                ada.Fill(ds);
                dataGridView1.DataSource = ds.Tables[0];
            }
        }
    }
}

```

```

    }
  }
}

```

Форма работы с поставщиками/Решение FormProvider.cs

```

namespace АРТЕКА
{
    public partial class FormProvider : Form
    {
        public FormProvider()
        {
            InitializeComponent();

            private void Workofprov_Click(object sender, EventArgs e)
            {
                SearchProvider pr2 = new SearchProvider();
                this.Hide();
                pr2.Show();
            }

            private void Providers_Click_1(object sender, EventArgs e)
            {
                DirProviders pr1 = new DirProviders();
                this.Hide();
                pr1.Show();
            }

            private void Exit_Click(object sender, EventArgs e)
            {
                Application.Exit();
            }

            private void Cancel_Click(object sender, EventArgs e)
            {
                this.Close();
                FormDirector mn = new FormDirector();
                mn.Show();
            }
        }
    }
}

```

Форма поставщиков/Решение DirProviders.cs

```

namespace АРТЕКА
{
    public partial class DirProviders : Form
    {
        public DirProviders()
        {
            InitializeComponent();
        }
    }
}

```

```

private void ПоставщикиBindingNavigatorSaveItem_Click(object sender,EventArgs e)
{
    this.Validate();
    this.поставщикиBindingSource.EndEdit();
    this.tableAdapterManager.UpdateAll(this.аптекаDataSet1);

}

private void DirProviders_Load(object sender, EventArgs e)
{
    //данная строка кода позволяет загрузить данные в таблицу
    "аптекаDataSet1.Поставщики".
    this.поставщикиTableAdapter.Fill(this.аптекаDataSet1.Поставщики);
}

private void Exit_Click(object sender, EventArgs e)
{
    Application.Exit();
}

private void Cancel_Click(object sender, EventArgs e)
{
    this.Close();
    FormProvider pr = new FormProvider();
    pr.Show();
}
}
}

```

Форма поиска поставщиков по препарату/Решение SearchProvider.cs

```

namespace АПТЕКА
{
    public partial class SearchProvider : Form
    {
        string strConn = FormEnter.strConn;
        SqlDataAdapter ada;
        DataSet ds;
        public SearchProvider()
        {
            InitializeComponent();
        }

        private void Button1_Click(object sender, EventArgs e)
        {
            using (SqlConnection conn = new SqlConnection(strConn))
            {
                conn.Open();
                SqlCommand comm = new SqlCommand("SELECT Препараты.Название,
                Препараты.Форма, Поставщики.Название as 'Поставщик', Поставщики.Адрес,
                Поставщики.Телефон, Поставки.[Дата поставки], Поставки.Количество FROM
                Препараты INNER JOIN(Поставщики INNER JOIN Поставки ON Поставщики.Код =

```

```

Поставки.[Код Поставщика]) ON Препараты.Код = Поставки.[Код Препарата] WHERE
Препараты.Название LIKE "%" + textBox1.Text.ToString() + "%", conn);
        ada = new SqlDataAdapter(comm);
        ds = new DataSet();
        ada.Fill(ds);
        dataGridView1.DataSource = ds.Tables[0];
    }
}

private void Exit_Click(object sender, EventArgs e)
{
    Application.Exit();
}

private void Cancel_Click(object sender, EventArgs e)
{
    this.Close();
    FormProvider pr = new FormProvider();
    pr.Show();
}
}
}

```

СПИСОК ИСТОЧНИКОВ

1. http://www.newart.ru/htm/flash/risovalka_42.php - рисование диаграммы прецедентов и ER-модели
2. Учебное пособие «Разработка приложений для работы с базами данных на языке С#», Составитель Иваньчева Т.А.