



МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технологический университет «СТАНКИН»
(ФГБОУ ВО МГТУ «СТАНКИН»)

Институт
информационных технологий

Кафедра
информационных систем

КУРСОВОЙ ПРОЕКТ

по дисциплине «**Управление данными**»
на тему: Проектирование базы данных аптеки

Студент
группа ИДБ-21-06

Руководитель
старший преподаватель


подпись

Виговская П. И.


подпись

Быстрикова В. А.

Москва 2023 г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
ГЛАВА 1. ПРОЕКТИРОВАНИЕ СТРУКТУРЫ БАЗЫ ДАННЫХ	4
1.1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	4
1.2. КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ	17
1.3. ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ	20
1.4. ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ	26
ГЛАВА 2. ОПИСАНИЕ ФУНКЦИОНИРОВАНИЯ БАЗЫ ДАННЫХ	35
2.1. НАЗНАЧЕНИЕ И ПЕРЕЧЕНЬ ФУНКЦИЙ БАЗЫ ДАННЫХ	35
2.2. ОПИСАНИЕ РАБОТЫ С БАЗОЙ ДАННЫХ	36
ЗАКЛЮЧЕНИЕ	61
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	63
ПРИЛОЖЕНИЕ А. SQL-ОПЕРАТОРЫ СОЗДАНИЯ ПРОГРАММНЫХ ОБЪЕКТОВ БД.....	65
ПРИЛОЖЕНИЕ Б. ЗАПОЛНЕНИЕ ТАБЛИЦ ДАННЫМИ.....	74
ПРИЛОЖЕНИЕ В. КОД ПРОГРАММЫ.....	80

ВВЕДЕНИЕ

Объектом исследования курсовой работы является аптека. Аптека – это учреждение, занимающееся продажей и, как правило, изготовлением лекарств и средств для ухода за здоровьем. Данная тема актуальна по ряду причин: сейчас во многих областях внедряется и активно используется цифровизация, а также потому что для обеспечения ведения информации о деятельности аптеки и оптимизации её деятельности необходима автоматизированная система, основанная на современной базе данных. Использование автоматизированной системы существенно сократит время поиска информации о препаратах и решит многие другие задачи.

Основной целью данного курсового проекта является изучение устройства аптеки и создание БД, хранящей всю информацию о ее деятельности, а также удобного пользовательского интерфейса для взаимодействия с БД.

Для выполнения курсовой работы необходимо провести анализ предметной области и выявить функции, которые выполняет аптека. На этапе концептуального проектирования для формирования функциональных требований нам необходимо выявить основное содержание базы данных без ориентации на какую-либо СУБД. Следующим шагом к выполнению курсового проекта является логическое проектирование. На данном этапе необходимо построить ER-диаграммы, где будет показано, как разные сущности связаны между собой в системе. Следующим этапом является физическое проектирование. На этом этапе рассматриваются основные отношения, организация файлов и индексов, предназначенных для обеспечения эффективного доступа к данным, а также все связанные с этим ограничения. После этого необходимо создать пользовательский интерфейс для взаимодействия сотрудников с базой данных.

ГЛАВА 1. ПРОЕКТИРОВАНИЕ СТРУКТУРЫ БАЗЫ ДАННЫХ

1.1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Предметную область можно определить как сферу человеческой деятельности, выделенную и описанную согласно установленным критериям.

В описываемое понятие должны входить сведения об ее элементах, явлениях, отношениях и процессах, отражающих различные аспекты этой деятельности.

Одна из первых задач, с решением которых сталкивается разработчик программной системы – это изучение, осмысление и анализ предметной области. Дело в том, что предметная область сильно влияет на все аспекты проекта: требования к системе, взаимодействие с пользователем, модель хранения данных, реализацию и т.д.

Анализ предметной области позволяет выделить ее сущности, определить первоначальные требования к функциональности и определить границы проекта [1].

В данной работе в качестве предметной области рассматривается деятельность аптеки.

Актуальность выполнения данной работы заключается в том, что в аптеках доступно большое количество различных препаратов, и необходимо иметь актуальную информацию о наличии и ценах на эти товары, требуется оптимизировать процесс заказа медикаментов и ведения учета запасов на складе для улучшения эффективности работы аптеки и снижения затрат, периодически появляются новые медикаменты на рынке, из-за чего требуются обновления базы данных. Кроме того, база данных аптеки может быть использована для мониторинга и анализа рынка медикаментов, что может помочь в принятии стратегических решений по улучшению бизнес-процессов и повышению прибыли. Регулярное обновление информации в базе данных позволяет избежать ошибок при продаже и закупке товаров, а также соблюдать

законодательные требования относительно ведения учета лекарств и предоставления соответствующих отчетов о продажах.

В деятельности аптеки под термином "товар" подразумеваются фармацевтические продукты, предназначенные для лечения заболеваний и поддержания здоровья. Это могут быть лекарственные препараты, биологически активные добавки, медицинские изделия, устройства для медицинских процедур, средства гигиены и т.д. Все товары, продающиеся в аптеках, должны соответствовать требованиям законодательства и проходить необходимую сертификацию.

Основными задачами, решаемыми в данной предметной области, являются:

- Учет товаров. Включает хранение информации обо всех лекарствах, которые продаются в аптеке, включая наименование, описание, дозировку, производителя и цену.
- Учет клиентов.
- Учет продаж. Дата продажи, проданные товары и кассовый чек.
- Оптимизация управления запасами. Планирование закупок лекарственных препаратов, создание эффективной стратегии управления запасами, исходя из анализа потребностей и динамики продаж.
- Учет персонала. Данные о сотрудниках аптеки, такие как их контактная информация, должностные обязанности и прочие данные.
- Статистический анализ. Проведение статистического анализа продаж, анализа прибыли, расчета КПД и формирования отчетов о продажах по определенному периоду.
- Мониторинг качества. Отслеживание сроков годности лекарств и контроля их качества.
- Контроль за исполнением рецептов. Учет рецептов на лекарственные препараты: регистрация принятых рецептов, контроль предъявления документов, проверка правильности выдачи лекарства, ведение электронных карт пациентов.

- Работа с поставщиками и производителями. Учет поставщиков, с которыми заключены договоры, выявления потребностей в новых поставках, а также контроля качества поступающих препаратов.

Существует множество программных продуктов, решающих задачи в данной области:

- конфигурация «Аптека» от компании «Простой софт»;
- программный продукт 1С:Аптека от компании 1С;
- центр решений БЭСТ-5.Аптека;
- система для аптек и аптечных сетей Стандарт-Н;
- решение «БИТ.АПТЕКА» от компании «Первый БИТ» и другие [2].

Такое программное обеспечение предназначено для упрощения и оптимизации работы аптечного предприятия. Оно позволяет автоматизировать все основные процессы, связанные с продажей медицинских товаров, учетом товаров на складе, учетом финансовых операций и контролем за сроками годности. Оно помогает значительно сократить время, необходимое для обработки заказов и составления отчетности. Кроме того, обеспечивает более точную и своевременную информацию о товарах на складе и о финансовых показателях, что позволяет принимать более обоснованные решения в управлении предприятием.

Программное обеспечение для автоматизации деятельности аптек является необходимым инструментом для эффективного управления бизнесом и повышения качества обслуживания клиентов.

В качестве первого программного продукта рассмотрим 1С:Аптека [3]. Отраслевое решение "1С:Аптека" предназначено для автоматизации розничных продаж лекарственных препаратов и сопутствующей продукции в аптеках и аптечных пунктах, ведения оперативного учета по сериям лекарственных препаратов, контроля фальсификации и сроков годности, правил ценообразования, предельных наценок на жизненно-необходимые и важные лекарственные препараты.

Среди отраслевых особенностей учета стоит выделить:

- Учет производства лекарственных форм при помощи документа "Изготовление лекарственных форм" (рис.1.1.1), в котором указываются рецептурная пропись, наименование производимой лекарственной формы, количество изготавливаемой лекарственной формы, составляющие (перечень лекарственных препаратов, идущих на изготовление лекарственной формы), сумма дополнительных расходов.

Изготовление лекарственных форм 00000000001 от 01.02.2017 15:49:05

Провести и закрыть

Номер: 00000000001 от: 01.02.2017 15:49:05

Рецептурная пропись: Рецептурная пропись 00000000001 от 0

Магазин: Аптека №1 Организация: ООО "Медикаменты"

Склад: Склад Аптека №1 Хоз. операция: Индивидуальное изготовление ЛС, в том числе

Лекарственная форма

Номенклатура: Молочко Видаля Количество: 0,500 Упаковка: Ед. изм.: шт

Характеристика: Доп. расходы: 100,00 Цена: 676,50 Сумма: 338,25

Товары	Комментарий	Артикул	Номенклатура	Характеристика	Количе...	Ед. изм.	Цена	Сумма
			Сера осажденная	65667, 28.02.2017	3,000	шт	66,00	198,00
			Спирт камфорный	66765, 20.02.2017	0,500	шт	25,00	12,50
			Раствор кислоты б...	565544, 21.04.2018	0,250	шт	66,00	16,50
			Спирт этиловый 96%	546445, 20.03.2017	0,250	шт	45,00	11,25

Ответственный: Борисов Федор Михайлович

Рис.1.1.1 Изготовление лекарственных форм

- Для каждой учетной позиции предоставляется возможность хранить данные о единице измерения, форме выпуска, дозировке, производителе, сертификате, фармакотерапевтической группе, международном непатентованном наименовании, наборах упаковок, ставке НДС, стране происхождения товара (рис.1.1.2).

Диазолин 0,1 № 10 (Позиция номенклатуры)

Наименование: Диазолин 0,1 № 10 Код: 00-00000002

Полное наименование: Диазолин 0,1 № 10

Вид номенклатуры: Лекарственные средства

Учетная информация: Товар

Тип номенклатуры: Товар

Характеристики: Индивидуальные для номенклатуры

Входит в группу: Антиаллергические средства

Артикул: 12054001

Ставка НДС: 10%

Набор упаковок: шт

Единица измерения: шт

Товарная группа: Лекарственные средства

Ценовая группа: ЖНВЛП

Вес (нетто): 0,000 кг

Характеристика номенклатуры: 65656, 31.08.2017

Номенклатура: Диазолин 0,1 № 10

Параметры серии

Дата изготовления: 31.08.2017 Срок хранения: 0 Срок годности: 31.08.2017

Серия производителя: 65656

Сертификат: Не использовать

Производитель: Производитель - 3М Медика

Наименование характеристики: 65656, 31.08.2017

Дополнительные реквизиты

Общие свойства характеристик

Страна происхождения: Страна происхождения - ЛАТВИЯ

Штрихкоды

Штрихкод	Тип штрих...	Упаковка	Владелец
2200000000019	EAN13		Диазолин 0,1 ...

Рис.1.1.2 Учёт данных товаров

- Возможность работы в режиме распределенных информационных баз (РИБ) с четким разделением документооборота по аптекам, где в главном узле РИБ консолидируется информация по всем аптекам сети.
- Используются ордерные схемы перемещения, реализации и поступления товаров на склады аптеки: перемещение в буферный список товаров, необходимых к принятию или отгрузке со склада. Также для перемещения товаров между аптеками предусмотрена безордерная схема работы: списание лекарственного препарата с остатков в одной аптеке и оприходование этого препарата в другой аптеке осуществляется в момент проведения документа "Перемещение товаров" (рис.1.1.3).

Демонстрационная база / Аптека №1 / Борисов Федор Михайлович / (ИС.Предприятие)

Начальная страница

Перемещение товаров

Перемещение товаров 00PT-000001 от 28.02.2017 17:36:03

Провести и закрыть

Распределение товаров

Номер: 00PT-000001 от 28.02.2017 17:36:03

Магазин отправитель: Аптека №1

Магазин получатель: Аптека №2

Склад отправитель: Отдел 1 (Аптека №1)

Склад получатель: Отдел 2 (Аптека №2)

Организация отправитель: ООО "Медикаменты"

Организация получатель: Предприниматель ИЧП

Товары (6)

N	Артикул	Номенклатура	Характеристика	Количество	Ед изм	Цена	Сумма	Заказ на перем
1	1500981	Супрастин 25мг та...	789-45, 29.01.2011	1,000	Шт	279,50	279,50	
2	528963	Уголь активирован...	2874, 20.09.2013	1,000	Шт	35,10	35,10	

Сумма товаров: 1 411,80 Сумма услуг: 0,00 Сумма: 1 411,80

Ответственный: Борисов Федор Михайлович

Рис.1.1.3 Перемещение товаров

- В системе поддерживаются различные способы управления розничными ценами в торговой сети. Центральный офис сети может назначать розничные цены как для всей сети в целом, так и для групп аптек или каждой аптеки отдельно. На отдельные группы номенклатуры розничные цены могут также назначаться менеджером аптеки.

- Специализированный интерфейс рабочего места первостольника (первостольник – фармацевт-провизор, который стоит за окошком-прилавком, называемым первым столом [4]) для оформления розничных продаж. Для повышения эффективности продаж сопутствующих товаров реализован механизм учета товаров, продаваемых совместно (рис.1.1.4).

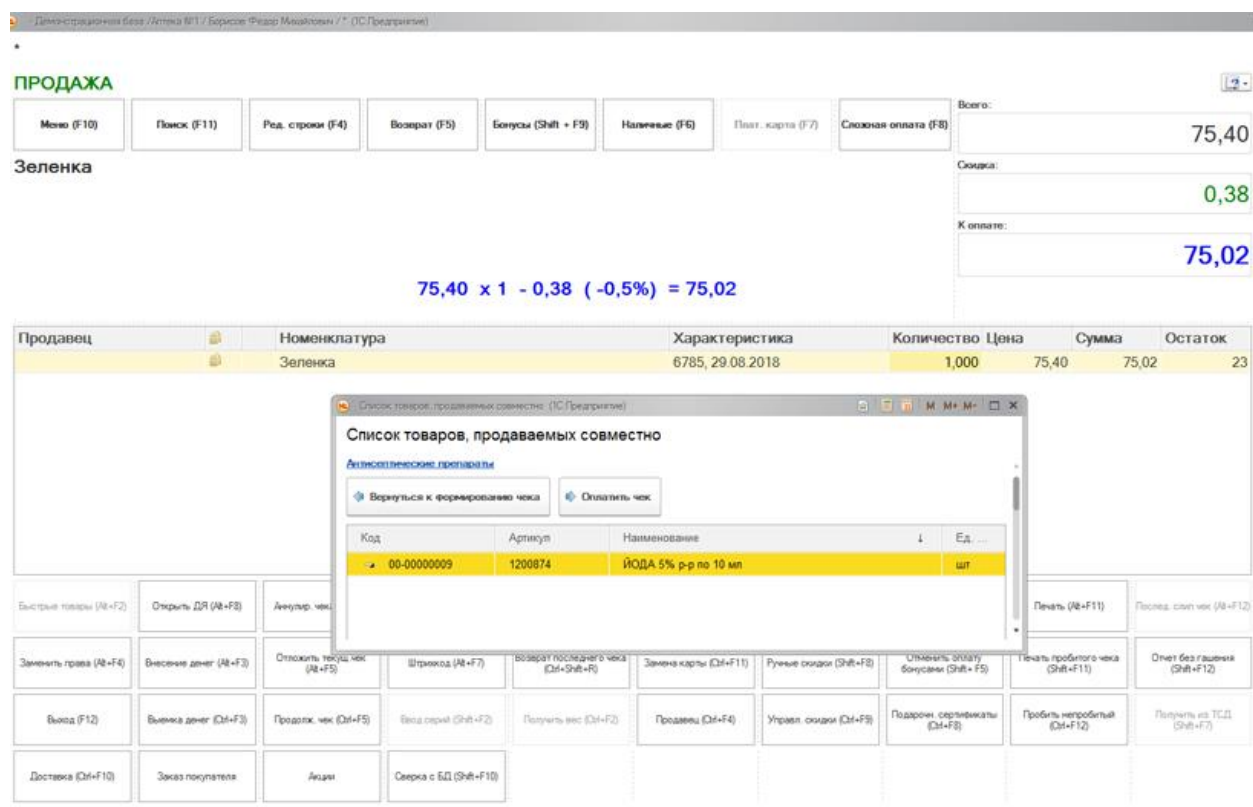


Рис.1.1.4 Учёт товаров, продаваемых совместно

- Для фиксации предварительной договоренности с покупателем о намерении приобрести товар реализована возможность создания заказа покупателя в режиме рабочего места кассира и продажа товара по заказу (рис.1.1.5).

Демонстрационная база / Аптека №1 / Борисов Федор Михайлович / * (1С:Предприятие)

ПРОДАЖА

Меню (F10) Поиск (F11) Ред. строки (F4) Возврат (F5) Бонусы (Shift + F9) Наличные (F6) Плат. карта (F7) Сложная оплата (F8) Всего: 0,00

Сумма: 0,00

Сумма: 0,00

Сумма: 0,00

Продавец

Формирование заказа покупателя *

Номер: от 01.02.2017 17:05:46

Контрагент: Розничный покупатель Магазин: Аптека №1

Дисконтная карта: Склад сборки: Склад Аптека №1

Продавец: Ковальков Н. Ф. Организация: ООО "Медикаменты"

Желаемая дата продажи: 01.02.2017

☐ Характеристики не обязательны к заполнению

Товары

Добавить Подобрать товары... Цены Скидки (наценки) Резерв Еще

N	Код	Резерв	Артикул	Номенклатура	Характеристика	Количество	Упаковка, Ед. изм.
1		✓	12054001	Диазолин 0,1 ...	3567859, 08.1...	1,000	шт
2		✓	1548971	Зеленка	6785, 29.08.2...	1,000	шт
3			1002541	Викасол ампл...	857421, 20.09...	1,000	шт
4		✓	12003247	Гликолат "С		1,000	шт

Итого

Сумма с НДС: 2 616,39 Сумма без скидки: 2 625,86

Сумма НДС: 374,82 Скидка (наценка): 9,47

Адрес доставки

Комментарий:

Печать Создать документ Закрыть

Быстрые товары (Alt+F2)

Зачесть права (Alt+F4)

Выход (F12)

Доставка (Ctrl+F10)

Заказ покупателя

Акции

Сверка с БД (Shift+F10)

Послед. счет-чек (Alt+F12)

Отчет без гашения (Shift+F12)

Получить из ТСОД (Shift+F7)

Рис.1.1.5 Продажа по заказу

- Для поддержки мультиканальности продаж, реализована интеграция с сайтами аптек на платформе "1С-Битрикс". Из конфигурации на сайт выгружается номенклатура, фото, цены, остатки и статусы обработанных заказов, с сайта в конфигурацию загружается информация о заказе, оплате заказа и о покупателе.
- Возможность формирования более 30 видов аналитических отчетов с разным уровнем группировки и детальными расшифровками по всем видам операций (рис.1.1.6).

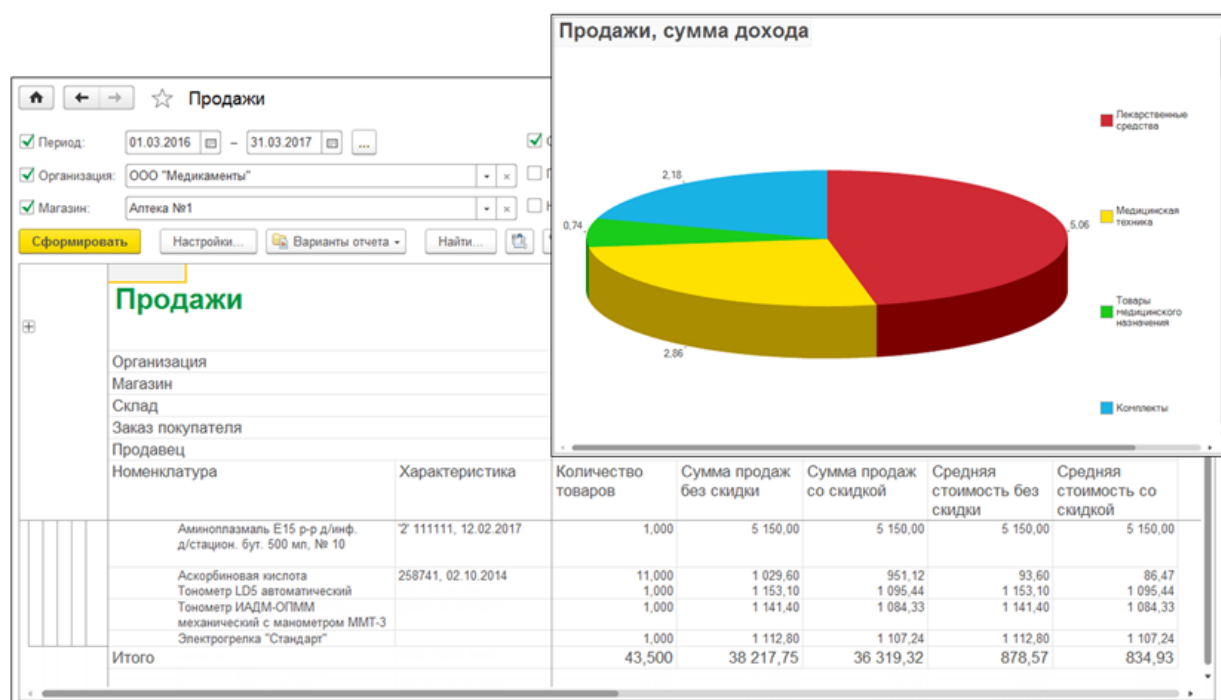


Рис.1.1.6 Аналитический отчет

В качестве второго программного продукта рассмотрим конфигурацию «Аптека» от компании «Простой софт» [5]. Конфигурация "Аптека" - предназначена для автоматизации аптек. Данная программа для аптеки позволит усовершенствовать процесс обслуживания клиентов благодаря ведению всестороннего учета. Имеется возможность ведения справочника товаров, лекарственных средств, болезней, учет заказов поставщикам, контроль поступления, списания товара, контроль остатков в режиме реального времени, ведение кассы предприятия, регистрация продаж и многое другое.

Функционал программного продукта включает в себя:

- Учет товаров. Ведение номенклатуры товаров аптеки со всеми необходимыми данными по ним (рис.1.1.7).

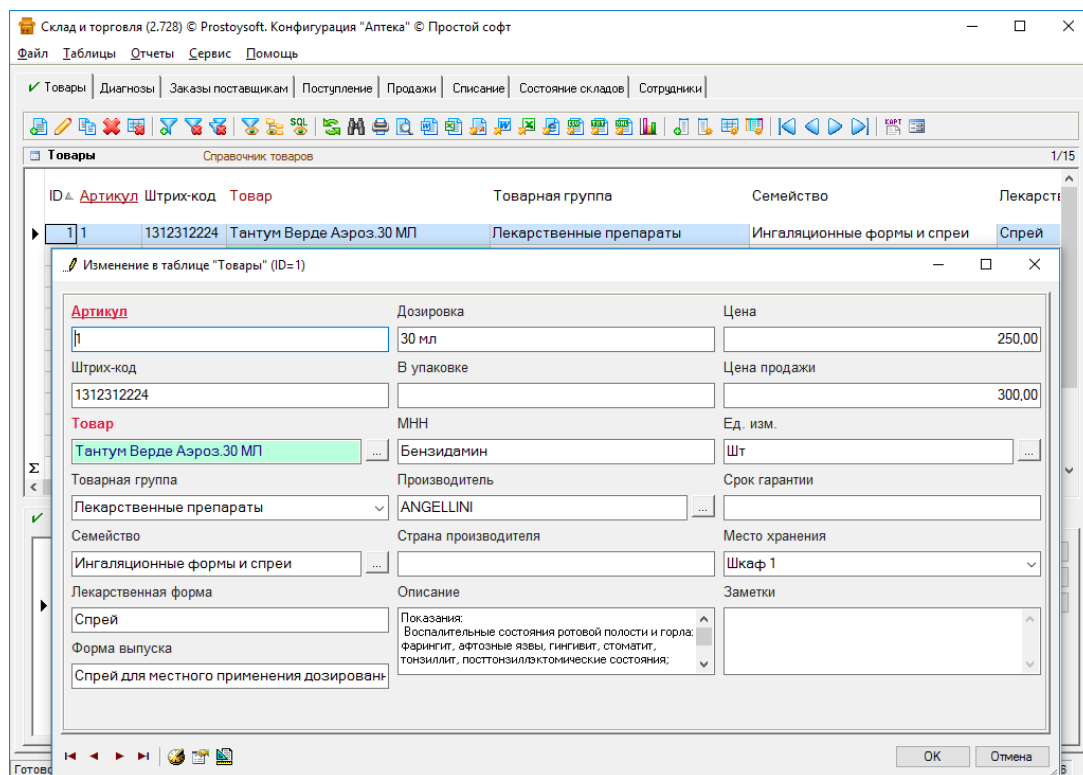


Рис.1.1.7 Учет товаров

- Работа с поставщиками. Регистрация заказов поставщикам, автоматический перенос номенклатуры заказа в поступление.
- Контроль состояния склада. Внутреннее перемещение, поступление, списание товара, контроль остатков в режиме реального времени (рис.1.1.8).

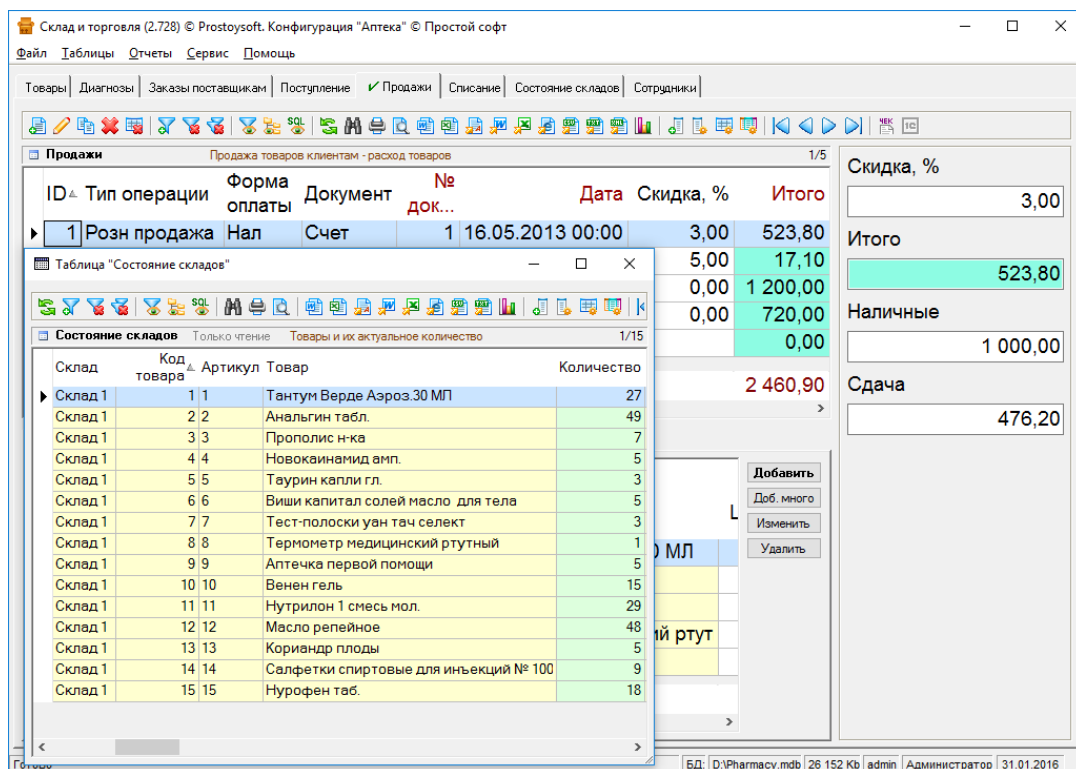


Рис.1.1.8 Состояние склада

- Учет клиентов. Данные о постоянных клиентах аптеки с их контактной информацией, учет скидок.
- Учет продаж (рис.1.1.9). Ведение кассы предприятия, печать чеков (рис.1.1.10), ценников.

Склад и торговля (2.728) © Prostoysoft. Конфигурация "Аптека" © Простой софт

Файл Таблицы Отчеты Сервис Помощь

Товары | Диагности | Заказы поставщикам | Поступление | **Продажи** | Списание | Состояние складов | Сотрудники

Продажи Продажа товаров клиентам - расход товаров 1/5 Скидка, %

Изменение в таблице "Продажи" (ID=1)

Продажа | Информация о продаже

Скидка, % Итого Наличные Сдача

3,00 523,80 1 000,00 476,20

Статус продажи Форма оплаты

Закрота Нал

Проданные товары (1/5)

ID	№	Код товара	Штрих-код	Артикул	Товар	Цена продажи	Количество	Добавить
1	1	1	1312312224	1	Тантум Верде Аэроз.30 МЛ	300,00	1	Доб. много
2	2	2	123324555	2	Анальгин табл.	12,00	1	Изменить
3	3	12	3123213	12	Масло репейное	60,00	1	Удалить
4	4	8	21331213	8	Термометр медицинский ртут	54,00	2	
5	5	3	423214424	3	Прополис н-ка	30,00	2	
Σ						5		

OK Отмена

Рис.1.1.9 Учет продаж

Склад и торговля (2.728) © Prostoysoft. Конфигурация "Аптека" © Простой софт

Файл Таблицы Отчеты Сервис Помощь

Товары | Диагности | Заказы поставщикам | Поступление | **Продажи** | Списание | Состояние складов | Сотрудники

Продажи Продажа товаров клиентам - расход товаров 1/5 Скидка, %

Изменение в таблице "Продажи" (ID=1)

Продажа | Информация о продаже

Скидка, % Итого Наличные Сдача

3,00 523,80 1 000,00 476,20

Статус продажи Форма оплаты

Закрота Нал

Проданные товары (1/5)

ID	№	Код товара	Штрих-код	Артикул	Товар	Цена продажи	Количество	Добавить
1	1	1	1312312224	1	Тантум Верде Аэроз.30 МЛ	300,00	1	Доб. много
2	2	2	123324555	2	Анальгин табл.	12,00	1	Изменить
3	3	12	3123213	12	Масло репейное	60,00	1	Удалить
4	4	8	21331213	8	Термометр медицинский ртут	54,00	2	
5	5	3	423214424	3	Прополис н-ка	30,00	2	
Σ						5		

ОPOS POS Printer

Печать чека

Принтер чеков: Posiflex Aura 6800/7000/8000 Денежный ящик: Posiflex CR-3100

☒ Мелкий шрифт

Шаблон чека:

ИНН 123456789112
[НАША ОРГАНИЗАЦИЯ]
Кассир: [CURRENTUSER]
ТЕСТОВЫЙ ЧЕК № [DocumentNumber]
[qdSalesProducts]
ИТОГО К ОПЛАТЕ: [SaleAmount]
Сохраняйте чек до получения товара
Спасибо за покупку!
[DATEANDTIME]

Печать Отмена

Готово

БД: D:\Pharmacy.mdb 26 152 Kb admin Администратор 31.01.2016

Рис.1.1.10 Печать чеков

- Хранение справочников. Возможность хранения и создания новых справочников ("ГРЛС", "МКБ-10", "Производители" и др.) (рис.1.11).

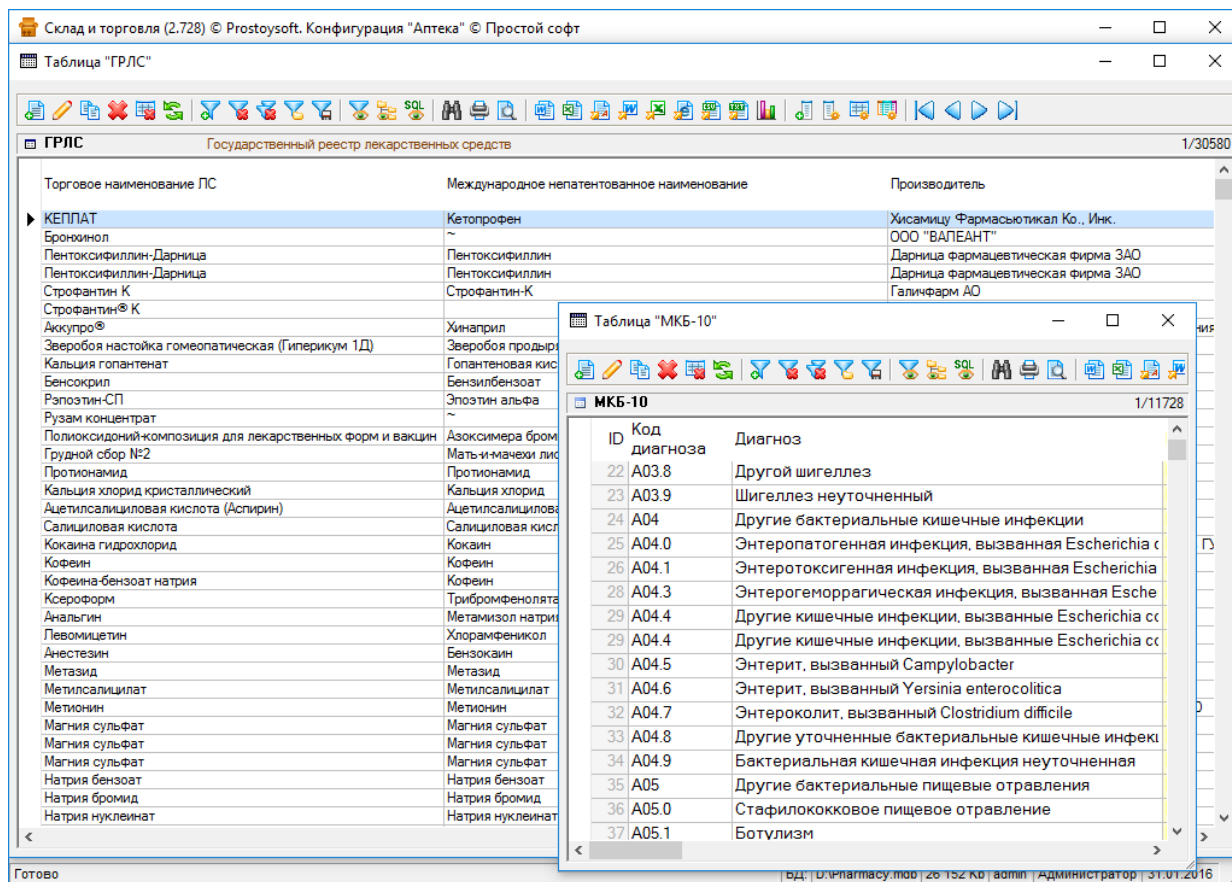


Рис.1.1.11 Справочники

- Учет сотрудников. Хранение информации о сотрудниках, настройка персональных прав доступа.
- Система напоминаний. Сообщения об актуальных задачах, поставленных пользователям на указанный срок.
- Набор стандартных отчетов с возможностью создания новых (рис.1.1.12).

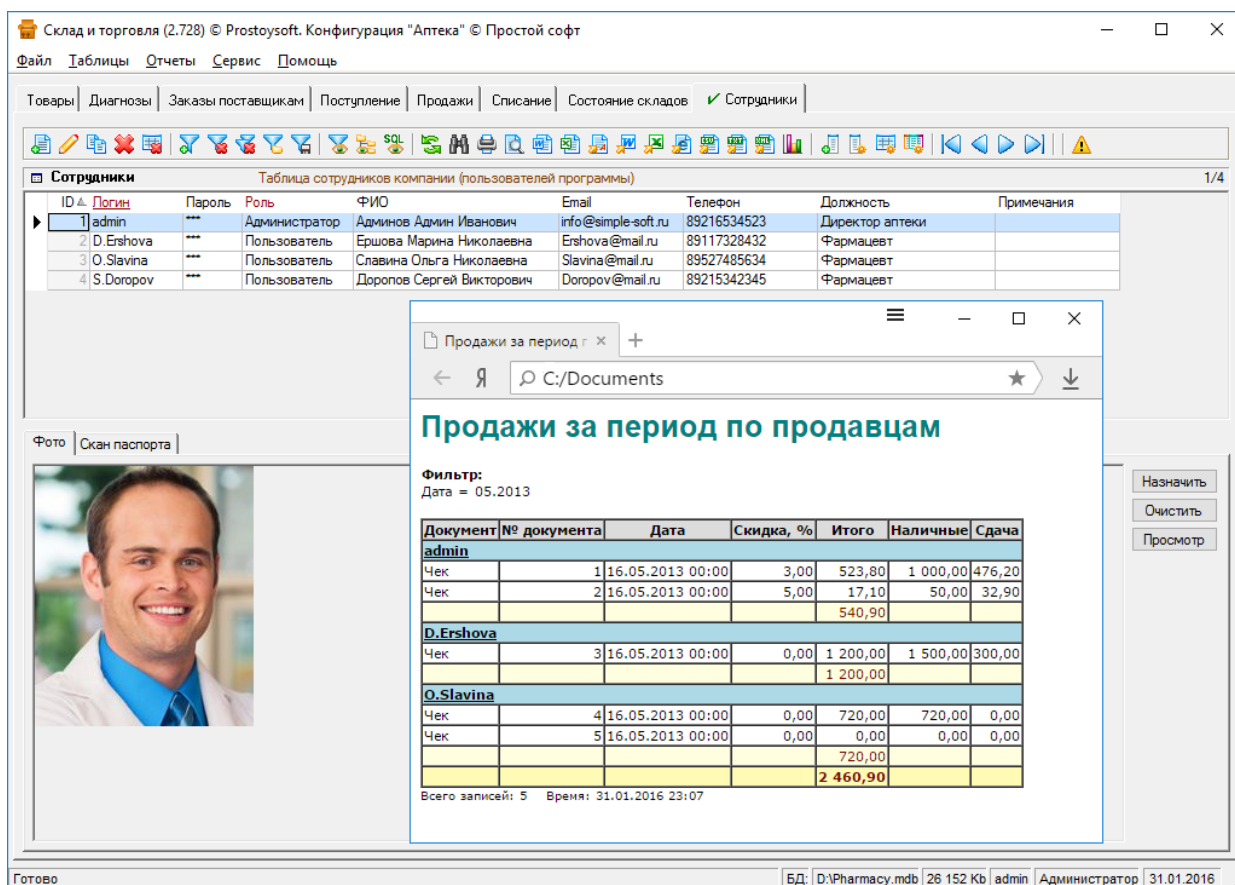


Рис.1.1.12 Создание отчёта

Конфигурация "Аптека" от компании "Простой софт" и программный продукт 1С:Аптека обладают некоторыми сходными возможностями:

- учет продажи лекарственных средств и медицинских изделий;
- учет остатков и приходов товаров;
- формирование документов: приходных накладных, счетов-фактур, актов на списание и перемещение товара;

- управление закупками;
- онлайн-режим работы для составления отчетов и управления данными.

Однако, есть также и различия между этими продуктами:

- Конфигурация «Аптека» от компании «Простой софт» имеет более простой интерфейс, что делает ее более понятной и легкой в использовании для персонала аптеки.

- Программный продукт 1С:Аптека имеет более широкий функционал, чем конфигурация «Аптека» от компании «Простой софт». Он позволяет

решать большой спектр задач и автоматизировать дополнительные процессы, такие как учет взаиморасчетов с поставщиками, управление скидками и акциями, анализ продаж и т.д.

- Программный продукт 1С:Аптека имеет высокую интеграцию с другими продуктами 1С, что позволяет управлять не только аптекой, но и другими областями бизнеса.

Выбор между конфигурацией «Аптека» от компании «Простой софт» и программным продуктом 1С:Аптека зависит от того, какие задачи необходимо решить, насколько сложный функционал требуется, и какой уровень интеграции необходим с другими программами.

После проведения анализа предметной области был выделен перечень функций, которые будут реализованы в данной работе:

- учет товаров;
- учет продаж;
- учет сотрудников;
- управление закупками;
- поиск товаров;
- подбор аналогов;
- учет остатков и приходов товаров;
- списание товаров с истекшим сроком годности.

1.2. КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ

Первым этапом проектирования базы данных является концептуальное проектирование. Концептуальное проектирование базы данных – это процесс создания модели используемой информации, не зависящей от любых физических аспектов ее представления [6]. Эта модель данных создается на основе информации, записанной в спецификациях требований пользователей. На этом этапе необходимо проанализировать задачи и построить диаграмму вариантов использования (ДВИ).

ДВИ нужна для формализации функциональных требований к системе с помощью действующих личностей и вариантов использования. Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества объектов, взаимодействующих с системой с помощью вариантов использования. Объектом называется любой объект, субъект или система, взаимодействующая с разрабатываемой системой извне. Вариант использования – это спецификация функций, которые система предоставляет объекту. Каждый вариант использования определяет некоторый набор действий, совершаемых системой при взаимодействии с объектом. При этом в модели никак не отражается то, каким образом будет реализован этот набор действий.

Анализ предметной области и этап концептуального проектирования связаны через функции, которые должно выполнять программное обеспечение. Анализ предметной области заключается в изучении требований пользователей и определении, какие данные должны храниться в базе данных, а также каким образом эти данные будут использоваться в рамках бизнес-процессов аптеки. На этапе концептуального проектирования необходимо определить функции, которые программа должна выполнять.

Составим словесное описание проектируемой БД.

По телефону потенциальные покупатели могут получить информацию о наличии товара, о ценах и характеристиках.

Покупатель, пришедший в аптеку, может получать у первостольника информацию о наличии товара, о ценах и характеристиках, кроме того, покупатель может получать фармацевтическое консультирование.

Для более оперативной работы с клиентами первостольнику и консультанту доступны поиск товаров и подбор аналогов.

Менеджер отслеживает расход товаров и состояние склада и принимает на этой основе решения о заказе новых партий у поставщиков, о продаже залежавшегося товара или списания товаров с истёкшим сроком годности. Также он ведёт учёт персонала, контролирует данные о сотрудниках аптеки.

С проектируемой системой будут взаимодействовать следующие действующие лица:

- первостольник,
- менеджер.

Первостольник должен быть фармацевтом (специалистом со средним фармацевтическим образованием) или провизором (специалистом с высшим фармацевтическим образованием), для менеджера это не обязательно.

В реальной жизни клиент аптеки может просматривать электронный каталог товаров. Однако в проектируемой системе такой возможности у него нет, поэтому покупатель не будет являться действующим лицом.

Рассмотрим варианты использования каждого из действующих лиц.

Первостольнику доступны следующие функции:

1. Выдача информации о характеристиках и ценах товаров.
2. Выдача информации о наличии товара.
3. Поиск по товарам и подбор аналогов.
4. Оформление продаж.

Менеджеру доступны следующие функции:

1. Учет товаров.
2. Выдача информации о наличии товара на складе.
3. Поиск по товарам.
4. Списание товаров.

5. Совершение закупок.
6. Учёт поставщиков.
7. Учёт сотрудников.
8. Контроль продаж.

Составим общую диаграмму вариантов использования (рис.1.2.1).

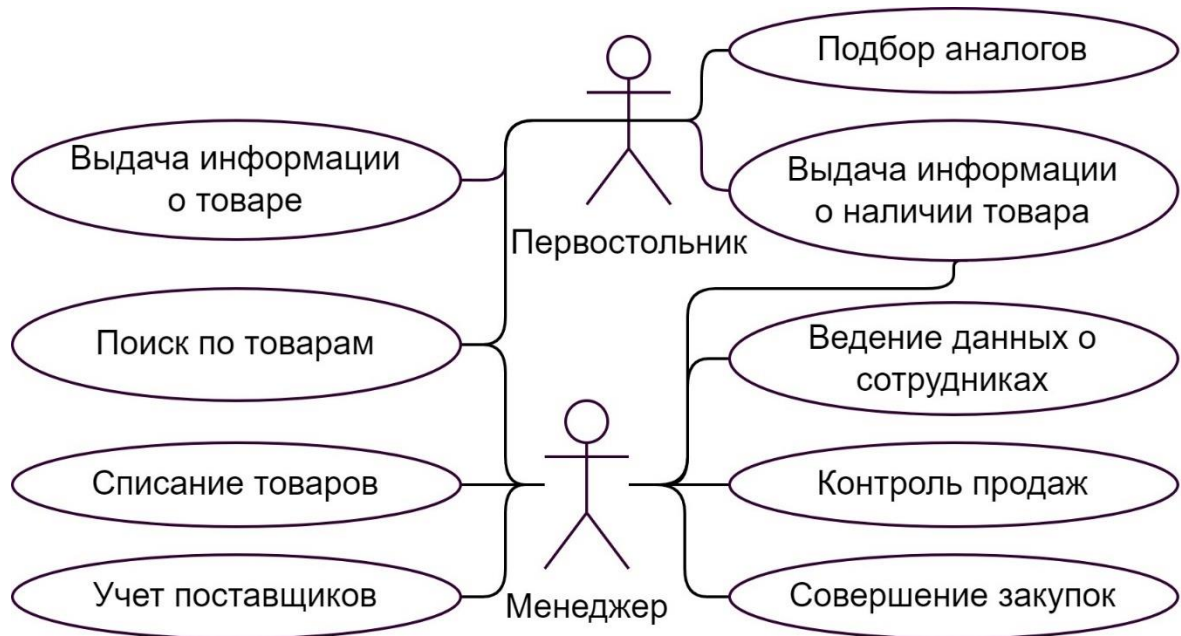


Рис.1.2.1. Диаграмма вариантов использования

В первом приближении для решения выделенных задач необходимо хранение данных о следующих объектах:

1. Продаваемые товары.
2. Регистрация поступления товаров на склад.
3. Истории продаж.
4. Сведения о сотрудниках.
5. Списание товаров с истёкшим сроком годности.

Необходимые данные можно классифицировать по частоте их изменения:

- условно-постоянные данные (например, характеристики продаваемых товаров);
- данные, которые оперативно обновляются при каждом решении задачи (какой товар, в каком количестве продан, когда поступил).

1.3. ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

Вторым этапом проектирования базы данных является логическое проектирование. Логическое проектирование базы данных – это преобразование требований к данным в структуры данных [7]. Цель логического проектирования – определение состава и структуры таблицы БД на основе результатов концептуального проектирования и проверка полученной модели с помощью методов нормализации.

В стадии логического проектирования входит:

- формирование отношений на основе логической модели данных;
- проверка отношений с использованием средств нормализации;
- определение ограничений целостности.

Для построения отношений используются диаграммы ER-типа сущность - связь. Основные ее определения:

- сущность – объект, информация о котором должна храниться в базе данных. Обозначается на ER-диаграмме прямоугольником, именуется существительным;
- связь – ассоциация между двумя сущностями, предполагает наличие общих атрибутов. Обозначается на ER-диаграмме ромбом, именуется глаголом.

Степень связи сущностей: 1:1, 1:M (M:1) или M:M. В общем случае между любой парой сущностей может быть задано любое количество связей. Графически степень связи указывается на линиях связи.

Класс принадлежности (КП) сущности может быть:

- обязательный (О) – в связи должен участвовать каждый экземпляр сущности (на диаграмме изображается дополнительным блоком с точкой внутри);
- необязательный (Н) – не все экземпляры сущности должны участвовать в связи (дополнительного блока нет).

На предыдущем этапе были выделены объекты, которые необходимо хранить в базе данных. Эти объекты становятся сущностями при ER-моделировании.

Построим ER-диаграммы всех сущностей и связей между ними.

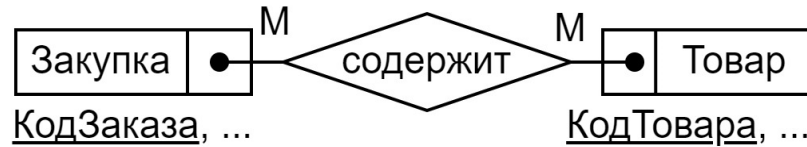


Рис.1.3.1. Связь «содержит»

Связь «содержит» (рис.1.3.1) имеет тип связи М:М, так как ЗАКУПКА имеет много товаров, а ТОВАР может содержаться в разных закупках. Сущность ЗАКУПКА имеет обязательный КП, так как каждая закупка имеет товары. Сущность ТОВАР имеет обязательный КП, так как каждый товар обязательно содержится в какой-либо закупке.

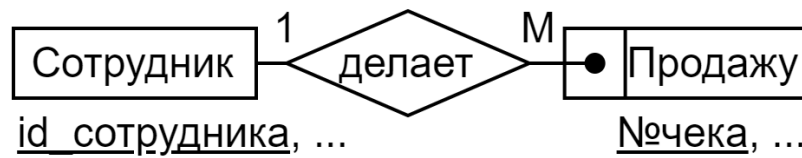


Рис.1.3.2. Связь «делает»

Связь «делает» (рис.1.3.2) имеет тип связи 1:М, так как СОТРУДНИК совершает разные продажи, а ПРОДАЖА может осуществляться только одним сотрудником. Сущность СОТРУДНИК имеет необязательный КП, так как не каждый сотрудник делает продажи. Сущность ПРОДАЖА имеет обязательный КП, так как каждая продажа совершается сотрудником.

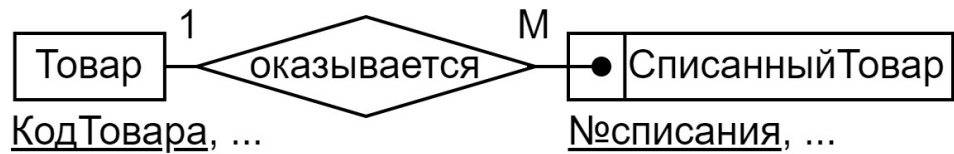


Рис.1.3.3. Связь «оказывается»

Связь «оказывается» (рис.1.3.3) имеет тип связи 1:М, так как ТОВАР может оказаться списанным много раз, а СПИСАННЫЙ ТОВАР является только одним товаром. Сущность ТОВАР имеет необязательный КП, так как не каждый товар оказывается списанным. Сущность СПИСАННЫЙ ТОВАР имеет обязательный КП, так как каждый списанный товар является товаром.



Рис.1.3.4. Связь «поставляет»

Связь «поставляет» (рис.1.3.4) имеет тип связи 1:M, так как ПОСТАВЩИК может поставлять много закупок, а ЗАКУПКА может быть только от одного поставщика. Сущность ПОСТАВЩИК имеет обязательный КП, так как у каждого поставщика, содержащегося в базе данных аптеки, будет иметься хотя бы одна закупка. Сущность ЗАКУПКА имеет обязательный КП, так как каждая закупка обязательно совершается каким-либо поставщиком.



Рис.1.3.5. Связь «совершает»

Связь «совершает» (рис.1.3.5) имеет тип связи 1:M, так как СОТРУДНИК может сделать много закупок, но ЗАКУПКА может быть совершена только одним сотрудником. Сущность СОТРУДНИК имеет необязательный КП, так как не каждый сотрудник занимается совершением закупок. Сущность ЗАКУПКА имеет обязательный КП, так как каждая закупка совершается каким-либо сотрудником.



Рис.1.3.6. Связь «обладает»

Связь «обладает» (рис.1.3.6) имеет тип связи M:1, так как ТОВАР имеет одну категорию, а у КАТЕГОРИИ может быть много товаров. Сущность ТОВАР имеет обязательный КП, так как у каждого товара обязательно есть категория. Сущность КАТЕГОРИЯ имеет обязательный КП, так как предполагается, что во всех имеющихся у аптеки категориях есть товары.



Рис.1.3.7. Связь «является»

Связь «обладает» (рис.1.3.7) имеет тип связи 1:1, так как ПРЕПАРАТ является одним товаром и ТОВАР является одним препаратом. Сущность ПРЕПАРАТ имеет обязательный КП, так как каждый препарат является товаром. Сущность ТОВАР имеет необязательный КП, так как не все товары являются препаратами.

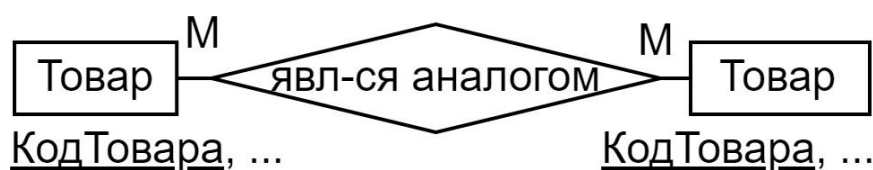


Рис.1.3.8. Связь «является аналогом»

Рекурсивная связь «является аналогом» (рис.1.3.8) имеет тип связи М:М, так как ТОВАР может являться аналогом нескольких товаров. Сущность ТОВАР имеет необязательный КП, так как не каждый товар может являться аналогом.

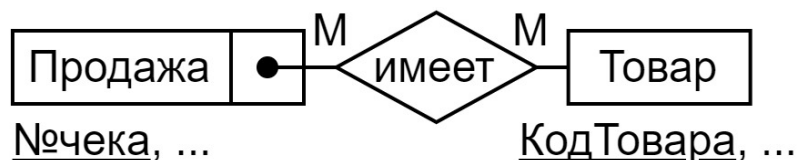


Рис.1.3.9. Связь «имеет»

Связь «имеет» (рис.1.3.9) имеет тип связи М:М, так как ПРОДАЖА может иметь много товаров и ТОВАР может иметь много продаж. Сущность ПРОДАЖА имеет обязательный КП, так как каждая продажа имеет товар. Сущность ТОВАР имеет необязательный КП, так как не все товары имеют продажи.

А) Сформируем набор предварительных отношений с указанием предполагаемого первичного ключа для каждого отношения.

Связь «содержит» (рис.1) удовлетворяет условиям правила 6, в соответствии с которым получается три отношения:

1. Товар (КодТовара, ...)
2. Закупка (КодЗаказа, ...)
3. СоставЗакупки (КодТовара, КодЗаказа ...)

Связь «делает» (рис.2) удовлетворяет условиям правила 4, в соответствии с которым получается два отношения:

1. Продажа (№чека, id_сотрудника, ...)
2. Сотрудник (id_сотрудника, ...)

Связь «оказывается» (рис.3) удовлетворяет условиям правила 4, в соответствии с которым получается два отношения:

1. Товар (КодТовара, ...)
2. СписанныйТовар (№списания, КодТовара, ...)

Связь «поставляет» (рис.4) удовлетворяет условиям правила 4, в соответствии с которым получается два отношения:

1. Поставщик (КодПоставщика, ...)
2. Закупка (КодЗаказа, КодПоставщика, ...)

Связь «совершает» (рис.5) удовлетворяет условиям правила 4, в соответствии с которым получается два отношения:

1. Сотрудник (id_сотрудника, ...)
2. Закупка (КодЗаказа, id_сотрудника, ...)

Связь «обладает» (рис.6) удовлетворяет условиям правила 4, в соответствии с которым получается два отношения:

1. Товар (КодТовара, КодКатегории, ...)
2. Категория (КодКатегории, ...)

Связь «является» (рис.7) удовлетворяет условиям правила 2, в соответствии с которым получается два отношения:

1. Товар (КодТовара, ...)
2. Препарат (КодПрепарата, КодТовара, ...)

Связь «является аналогом» (рис.8) удовлетворяет условиям правила 6, в соответствии с которым получается два отношения:

1. Товар (КодТовара, ...)

2. Аналог (КодТовара, КодТовараАналога)

Связь «имеет» (рис.9) удовлетворяет условиям правила 6, в соответствии с которым получается три отношения:

1. Товар (КодТовара, ...)
2. Продажа (№чека, ...)
3. СоставПродажи (№чека, КодТовара, ...)

Б) Добавим неключевые атрибуты в каждое из предварительных отношений с условием, чтобы отношения отвечали требованиям третьей нормальной формы.

Товар (КодТовара, КодКатегории, Марка, Название, Цена, МинимальныйЗапас, КоличествоВНаличии)

Препарат (КодПрепарата, КодТовара, ФармГруппа, СрокГодности, Примечания, Действие, ДействующееВещество, ФормаВыпуска, Дозировка, Количество, Объем, Состав, Страна, Производитель)

Категория (КодКатегории, Название)

Сотрудник (id_сотрудника, ФИО, Должность, ЭлектроннаяПочта, РабочийТелефон)

Продажа (№чека, id_сотрудника, ДатаПродажи)

Поставщик (КодПоставщика, НазваниеПоставщика, КонтактноеЛицо, Страна, Город, Адрес, Телефон, УсловияОплаты)

Закупка (КодЗаказа, КодПоставщика, id_сотрудника, ДатаИсполнения, СуммарнаяСтоимость)

СоставЗакупки (КодЗаказа, КодТовара, ГоденДо, ЦенаЕдиницы, Количество)

СписанныйТовар (№списания, КодТовара, Количество, ДатаСписания, ПричинаСписания)

СоставПродажи (№чека, КодТовара, Количество, ОбщаяСтоимость)

Аналог (КодТовара, КодТовараАналога)

1.4. ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

Третьим этапом проектирования базы данных является физическое проектирование. Физическое проектирование базы данных – это процесс подготовки описания реализации базы данных на вторичных запоминающих устройствах; на этом этапе рассматриваются основные отношения, организация файлов и индексов, предназначенных для обеспечения эффективного доступа к данным, а также все связанные с этим ограничения целостности и средства защиты [6].

Физическое проектирование является последним этапом создания проекта базы данных, при выполнении которого принимаются решения о способах реализации разрабатываемой базы данных. Во время предыдущего этапа проектирования была определена логическая структура базы данных, которая описывает отношения и ограничения в рассматриваемой прикладной области. Хотя эта структура не зависит от конкретной целевой СУБД, она создается с учетом выбранной модели хранения данных, например реляционной, сетевой или иерархической. Приступая к физическому проектированию базы данных, необходимо выбрать СУБД. Поэтому физическое проектирование неразрывно связано с конкретной СУБД. Между логическим и физическим проектированием существует постоянная обратная связь, так как решения, принимаемые на этапе физического проектирования с целью повышения производительности системы, способны повлиять на структуру логической модели данных.

Основной целью физического проектирования базы данных является описание способа физической реализации логического проекта базы данных.

В качестве СУБД для проектирования базы данных выбран SQL Server Management Studio – это бесплатная графическая среда, включающая набор инструментов для разработки сценариев на T-SQL и управления инфраструктурой Microsoft SQL Server. Среда SQL Server Management Studio – это основной, стандартный и полнофункциональный инструмент для работы

с Microsoft SQL Server, который предназначен как для разработчиков, так и для администраторов SQL Server [8].

Так как в задачи входит полное сопровождение Microsoft SQL Server, начиная от создания баз данных, написания SQL запросов, создания хранимых процедур и функций, и заканчивая администрированием SQL Server, включая управление безопасностью, то среда SQL Server Management Studio отлично подходит в качестве основного инструмента.

На основе логического проектирования, где были построены ER-диаграммы, а также сформированы отношения и добавлены атрибуты, необходимо составить структуры таблиц, с помощью которых можно будет физически спроектировать базу данных. Для создания БД необходимо составить структуры вышеперечисленных таблиц. Необходимо установить типы данных, возможность оставить поля незаполненным, назначить первичные и внешние ключи, установить ограничения, значения по умолчанию, ссылки на другие таблицы.

В таблице 1 представлена структура таблицы «Товар», в приложении Б (рис. Б.1) содержатся данные таблицы «Товар».

Таблица 1

Требования к структуре таблицы «Товар»

Столбец	Тип данных	Ноль?	Ключ	По умолчанию	Ограничение	Ссылка
КодТовара	char(13)	нет	Первичный			
КодКатегории	tinyint	нет	Внешний			Категория (КодКатегории)
Марка	varchar(30)	нет				
Название	varchar(170)	нет				
Цена	money	нет			≥ 0	
Минимальный Запас	smallint	да				
КоличествоВ Наличии	smallint	нет		0	≥ 0	

В таблице 2 представлена структура таблицы «Препарат», в приложении Б (рис. Б.2) содержатся данные таблицы «Препарат».

Таблица 2

Требования к структуре таблицы «Препарат»

Столбец	Тип данных	Ноль?	Ключ	По умолчанию	Ссылка
КодПрепарата	smallint	нет	Первичный		
КодТовара	char(13)	нет	Внешний		Товар (КодТовара)
ФармГруппа	tinyint	нет	Внешний		Категория (КодКатегории)
СрокГодности	varchar(20)	нет		"Более 12 месяцев"	
Примечания	varchar(150)	да			
Действие	varchar(50)	да			
Действующее Вещество	varchar(70)	нет			
ФормаВыпуска	varchar(80)	нет			
Дозировка	varchar(50)	да			
Количество	tinyint	да			
Объем	varchar(15)	да			
Состав	varchar(400)	нет			
Страна	varchar(30)	да			
Производитель	varchar(70)	нет			

В таблице 3 представлена структура таблицы «Категория», в приложении Б (рис. Б.3) содержатся данные таблицы «Категория».

Таблица 3

Требования к структуре таблицы «Категория»

Столбец	Тип данных	Ноль?	Ключ	По умолчанию	Ограничение	Ссылка
КодКатегории	tinyint	нет	Первичный			
Название	varchar(30)	нет				

В таблице 4 представлена структура таблицы «Сотрудник», в приложении Б (рис. Б.4) содержатся данные таблицы «Сотрудник».

Таблица 4

Требования к структуре таблицы «Сотрудник»

Столбец	Тип данных	Нуль?	Ключ	По умолчанию	Ограничение
id_сотрудника	smallint	нет	Первичный		
ФИО	varchar(60)	нет			
Должность	varchar(40)	нет			
ЭлектроннаяПочта	varchar(50)	да			
РабочийТелефон	varchar(15)	да			уникальный

В таблице 5 представлена структура таблицы «Продажа», в приложении Б (рис. Б.5) содержатся данные таблицы «Продажа».

Таблица 5

Требования к структуре таблицы «Продажа»

Столбец	Тип данных	Нуль?	Ключ	По умолчанию	Ссылка
№чека	smallint	нет	Первичный		
id_сотрудника	smallint	нет	Внешний		Сотрудник (id_сотрудника)
ДатаПродажи	date	нет		Текущая дата	

В таблице 6 представлена структура таблицы «Поставщик», в приложении Б (рис. Б.6) содержатся данные таблицы «Поставщик».

Таблица 6

Требования к структуре таблицы «Поставщик»

Столбец	Тип данных	Нуль?	Ключ	По умолчанию	Ограничение
КодПоставщика	smallint	нет	Первичный		
НазваниеПоставщика	varchar(20)	нет			
КонтактноеЛицо	varchar(50)	нет			
Страна	varchar(30)	нет		"Россия"	
Город	varchar(30)	нет			
Адрес	varchar(100)	нет			
Телефон	varchar(20)	нет			
УсловияОплаты	varchar(100)	да			

В таблице 7 представлена структура таблицы «Закупка», в приложении Б (рис. Б.7) содержатся данные таблицы «Закупка».

Таблица 7

Требования к структуре таблицы «Закупка»

Столбец	Тип данных	Ноль?	Ключ	По умолчанию	Ограничение	Ссылка
КодЗаказа	int	нет	Первичный			
КодПоставщика	smallint	нет	Внешний			Поставщик (КодПоставщика)
id_сотрудника	smallint	нет	Внешний			Сотрудник (id сотрудника)
ДатаИсполнения	date	нет		Текущая дата		
СуммарнаяСтоимость	money	нет		0	>=0	

В таблице 8 представлена структура таблицы «СоставЗакупки», в приложении Б (рис. Б.8) содержатся данные таблицы «СоставЗакупки».

Таблица 8

Требования к структуре таблицы «СоставЗакупки»

Столбец	Тип данных	Ноль?	Ключ	По умолчанию	Ограничение	Ссылка
КодЗаказа	int	нет	Первичный			Закупка (КодЗаказа)
КодТовара	char(13)	нет	Первичный			Товар (КодТовара)
ГоденДо	date	нет				
ЦенаЕдиницы	money	нет			>=0	
Количество	smallint	нет		0	>=0	

В таблице 9 представлена структура таблицы «СписанныйТовар», в приложении Б (рис. Б.9) содержатся данные таблицы «СписанныйТовар».

Таблица 9

Требования к структуре таблицы «СписанныйТовар»

Столбец	Тип данных	Ноль?	Ключ	По умолчанию	Ограничение	Ссылка
№списания	smallint	нет	Первичный			
КодТовара	char(13)	нет	Внешний			Товар (КодТовара)
Кол-во	smallint	нет		1	>=1	
Дата Списания	date	нет		Текущая дата		
Причина Списания	varchar(60)	нет		“Истекает срок годности”		

В таблице 10 представлена структура таблицы «СоставПродажи», в приложении Б (рис. Б.10) содержатся данные таблицы «СоставПродажи».

Таблица 10

Требования к структуре таблицы «СоставПродажи»

Столбец	Тип данных	Ноль?	Ключ	По умолчанию	Ограничение	Ссылка
№чека	smallint	нет	Первичный			Продажа(№чека)
КодТовара	char(13)	нет	Первичный			Товар (КодТовара)
Кол-во	smallint	нет		1	>=1	
Общая Стоимость	money	нет				

В таблице 11 представлена структура таблицы «Аналог», в приложении Б (рис. Б.11) содержатся данные таблицы «Аналог».

Таблица 11

Требования к структуре таблицы «Аналог»

Столбец	Тип данных	Ноль?	Ключ	По умолчанию	Ссылка
КодТовара	char(13)	нет	Первичный		Товар(КодТовара)
КодТовара Аналога	char(13)	нет	Первичный		Товар(КодТовара)

После создания таблиц с помощью SQL Server Manegement Studio [9] построена схема данных (рис.1.4.1).

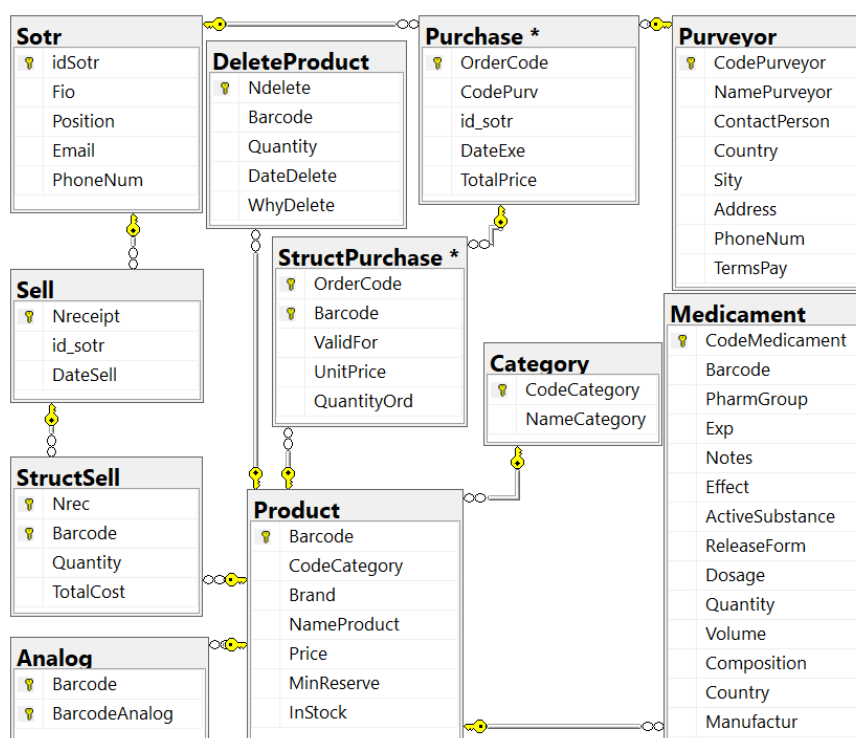


Рис.1.4.1 Схема данных

Также база данных имеет хранимые процедуры, функции и представления, необходимые для решения задач проектируемого приложения.

Представление deleteProductForWF (Приложение А, п.1) требуется для корректного отображения списанных товаров, основано на таблицах DeleteProduct и Product. Представление deleteProductForWF, в отличие от таблицы DeleteProduct, отображает не только штрихкод товара, но и его бренд и название.

Представление tableProduct (Приложение А, п.2) требуется для корректного отображения таблицы товаров, основано на таблицах Product и Category. Представление tableProduct, в отличие от таблицы Product, отображает категории товара, а не их коды.

Представление viewPurchase (Приложение А, п.3) требуется для отображения закупленных товаров, основано на таблицах Purchase, Purveyor, Sotr, StructPurchase, Product. Представление tableProduct отображает всю информацию о закупках товаров.

Представление viewSell (Приложение А, п.4) требуется для отображения проданных товаров, основано на таблицах Sell, StructSell, Sotr, Product. Представление viewSell отображает всю информацию о продажах товаров.

Скалярная функция checkBarcode (Приложение А, п.5) требуется для проверки существования штрихкода в таблице Product.

Скалярная функция checkStock (Приложение А, п.6) требуется для получения по штрихкоду количества товара в наличии из таблицы Product.

Скалярная функция getProductInfo (Приложение А, п.7) требуется для получения по штрихкоду бренда, названия и цены товара из таблицы Product.

Табличная функция getPurveyor (Приложение А, п.8) требуется для получения всех названий поставщиков из таблицы Purveyor.

Табличная функция findAnalog (Приложение А, п.9) требуется для получения по штрихкоду аналогов товара из таблицы findAnalog. Она выводит все характеристики из таблицы Product, кроме минимального запаса

и количества в наличии, для товаров, которые является аналогом товара, для которого они ищутся.

Табличная функция `seeDescriptionOfMedicament` (Приложение А, п.10) требуется для получения по штрихкоду характеристик медикаментов из таблицы `Medicament`.

Хранимая процедура `addMedicament` (Приложение А, п.11) требуется для добавления новых характеристик медикаментов в таблицу `Medicament`.

Хранимая процедура `addNewProduct` (Приложение А, п.12) требуется для добавления новых товаров в таблицу `Product`.

Хранимая процедура `addNewSotr` (Приложение А, п.13) требуется для добавления новых сотрудников в таблицу `Sotr`.

Хранимая процедура `addPurchase` (Приложение А, п.14) требуется для совершения закупок. Входными параметрами являются штрихкод товара, количество закупаемого товара, поставщик и сотрудник, совершающий закупку. В теле процедуры сначала по названию поставщика берется его код из таблицы `Purveyor`, затем проверяется существования в базе данных выбранного поставщика, и если такого поставщика нет, то выполнение процедуры прерывается. Затем по имени сотрудника берется его код из таблицы `Sotr`, по штрихкоду товара берется его цена, по этой цене и количеству закупаемого товара высчитывается суммарная стоимость. Номер заказа высчитывается автоматически: из таблицы `StructPurchase` берется максимальный номер заказа и прибавляется единица. Далее в таблицы `StructPurchase` и `Purchase` добавляются новые кортежи со значениями атрибутов, введенными и рассчитанными ранее. В самом конце обновляется количество товара в наличии в таблице `Product`.

Хранимая процедура `addPurveyor` (Приложение А, п.15) выполняет добавление новых поставщиков в таблицу `Purveyor`.

Хранимая процедура `addStructSell` (Приложение А, п.16) требуется для совершения продаж. Входными параметрами являются штрихкод товара, количество продаваемого товара и сотрудник, продающий товар. Процедура

начинается с проверки, есть ли продаваемый товар в наличии. Если товар есть в наличии, то процедура проверяет, достаточно ли товара для продажи в указанном количестве. Если товара достаточно, процедура продолжает выполнение. В теле процедуры выполняется несколько SELECT-запросов для получения идентификатора сотрудника, цены товара и вычисления общей стоимости продажи. Далее процедура генерирует новый номер чека продажи, учитывая наибольший номер из существующих чеков, и добавляет новые записи в таблицы "Sell" и "StructSell" со значениями атрибутов, введенными и рассчитанными ранее. В конце процедуры текущий остаток товара в таблице "Product" обновляется путем вычитания проданного количества товара.

Хранимая процедура authenticationUser (Приложение А, п.17) требуется для аутентификации пользователей. Входными параметрами являются ФИО сотрудника и его номер телефона. В теле процедуры определяется должность сотрудника по введенным данным.

Хранимая процедура retireSotr (Приложение А, п.18) требуется для увольнения сотрудников. Входным параметром является идентификатор сотрудника. В теле процедуры происходит обновление значения атрибута должности (добавление «Уволен» в начало) по идентификатору сотрудника.

Хранимая процедура writeOffProduct (Приложение А, п.19) требуется для списания товаров. Входными параметрами являются штрихкод товара, количество товара, которое нужно списать, и причина списания товара. В теле процедуры высчитывается количество товара в наличии и проверяется, что количество товара, которое нужно списать, не больше, чем количество товара в наличии. Если проверка пройдена, то таблица DeleteProduct пополняется записью о списании товара.

ГЛАВА 2. ОПИСАНИЕ ФУНКЦИОНИРОВАНИЯ БАЗЫ ДАННЫХ

2.1. НАЗНАЧЕНИЕ И ПЕРЕЧЕНЬ ФУНКЦИЙ БАЗЫ ДАННЫХ

Основной задачей курсового проекта является реализация удобного и понятного интерфейса для поиска, удаления, добавления и изменения информации в БД. Программный продукт должен автоматизировать основную часть работы аптеки. Работать с базой могут все сотрудники, однако функционал разделен по должностям для предотвращения деятельности сотрудников вне своих областей ответственности. В работе представлено два пользователя: первостольник и менеджер, для них необходимы свои интерфейсы для взаимодействия с БД.

Первостольник имеет такие функции, как:

- Поиск по товарам.
- Получения информации о ценах и количестве товара в наличии.
- Оформление продаж.
- Подбор аналогов.
- Просмотр характеристик препаратов.

Менеджер имеет такие функции, как:

- Добавление и увольнение сотрудников.
- Совершение закупок.
- Получение информации о наличии товара в аптеке.
- Списание товаров.
- Поиск по товарам и получение информации о них.
- Контроль продаж.
- Учёт поставщиков.

Пользовательский интерфейс разрабатывался с помощью Windows Forms [10] в среде разработки Visual Studio [11] на языке программирования C# [12].

2.2. ОПИСАНИЕ РАБОТЫ С БАЗОЙ ДАННЫХ

В начале работы для доступа к функционалу сотрудник должен авторизоваться (рис 2.2.1). После ввода логина и пароля требуется нажать кнопку «Войти». Код формы представлен в приложении В, п.1.

Рис.2.2.1 Вход в приложение

При попытке входа с некорректными данными высветится окно с текстом ошибки и счётчиком, отвечающим за количество доступных попыток входа в приложение (рис.2.2.2).

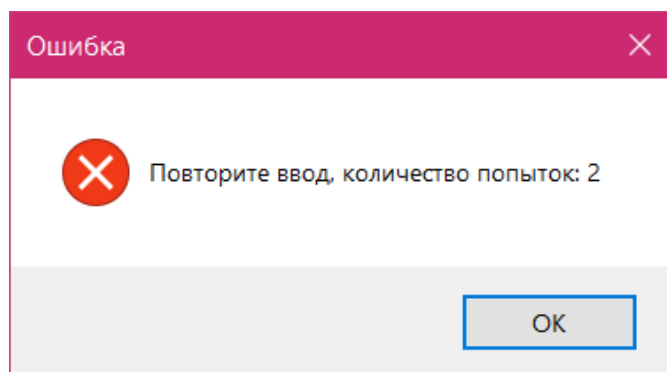


Рис.2.2.2 Окно ошибки при авторизации

Когда данный счетчик достигнет нуля, кнопка «Войти» станет недоступной (рис.2.2.3).

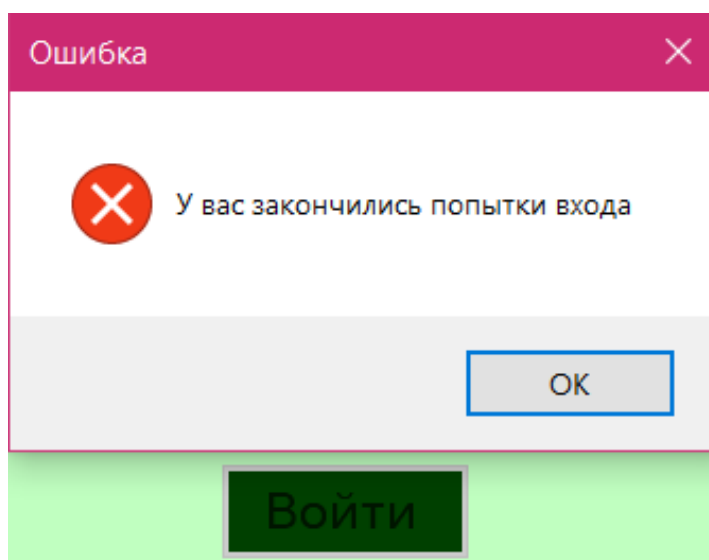


Рис.2.2.3 Блокировка кнопки

При успешной аутентификации срабатывает хранимая процедура «authenticationUser» (приложение А, п.1), и можно попасть на форму первостольника или менеджера.

Рассмотрим форму первостольника (рис.2.2.4). Код формы представлен в приложении В, п.2.

Штрихкод	Категория	Бренд	Название	Цена	Мин.запас	В наличии
0023456789004	Медицинские п...	Armed	концентратор к...	26289,0000	1	21
0023456789031	Медицинские п...	AMRUS	Термометр мед...	1250,0000	20	30
0023456789071	Гигиена	Б.Браун Медик...	Эпиген интим 0...	1902,0000	5	35
0023456789076	Медицинские и...	SFM HOSPITAL	Шприц 50 мл 3-...	862,0000	50	84
0123456789009	Медицинские и...	DERMAGRIP	Перчатки смот...	1095,0000	30	88
0123456789036	Гигиена	Amga	диски ватные 1...	116,0000	30	84
0723456789023	Обезболивающ...	Реневал	Парацетамол	50,0000	40	262
0812345678958	Ортопедически...	КОМФОРТ	Корректор оса...	1250,0000	5	30
1000987654325	Зрение	Maxima	55 comfort + ко...	941,0000	20	3
1001234567898	Зрение	OPTICLEAN	Opticlean салф...	42,0000	10	14
1009876543213	Зрение	Гельтек-Медика	Блефаросалфе...	505,0000	5	29
1123456789014	Медицинские п...	OMRON	Термометр омг...	563,0000	30	109
1123456789019	Гигиена	Kleenex	антибактериал...	174,0000	20	58
1123456789050	Медицинские и...	INFECTA	Катетер фалес...	500,0000	10	20

Рис.2.2.4 Интерфейс первостольника

Здесь первостольник может просматривать таблицу с товарами, где указаны их цены и количество в наличии. Таблица товаров выводится с

помощью представления «tableProduct» (Приложение А, п.2). Можно производить поиск по названию, бренду или категории товара: в поле вводятся необходимые данные, при нажатии на кнопку «Найти» осуществляется поиск по всем возможным совпадениям (рис.2.2.5). Вернуть изначальный вид таблицы можно с помощью кнопки «Сбросить», при ее нажатии также очищается поле ввода.

The screenshot shows the 'Первостольник' application window. At the top is a table with search results. Below the table is a search form with a text input field containing 'ингалятор' and two buttons: 'Найти' and 'Сбросить'. Below the search form is a section titled 'Оформление продажи' (Sale Form) with fields for 'Штрихкод' (Barcode), 'Количество' (Quantity), and 'Продал' (Sold by), along with buttons for 'Оформить продажу' (Complete Sale), 'Назад' (Back), 'Описание препаратов' (Drug Description), and 'Подбор аналогов' (Select Analogs).

Штрихкод	Категория	Бренд	Название	Цена	Мин.запас	В наличии
1123456789091	Медицинские п...	AND	Ингалятор CN-...	2535,0000	5	21
2223456789073	Медицинские п...	b well	Ингалятор wп-1...	2225,0000	5	13
3423456789052	Медицинские и...	ОСНОВНОЙ ЭП...	ингалятор-бал...	457,0000	8	10
4423456789030	Медицинские п...	AMRUS	Принадлежнос...	475,0000	5	54

Введите название, бренд или категорию товара:

Оформление продажи

Штрихкод:

Количество:

Продал:

Рис.2.2.5 Поиск по товарам

Первостольник может оформлять продажи товаров. Для этого требуется заполнить поля «Штрихкод» и «Количество» и нажать кнопку «Оформить продажу» (рис.2.2.6). При нажатии на кнопку срабатывает хранимая процедура «addStructSell» (приложение А, п.3). Поле сотрудника, продающего товар, заполнять не требуется, так как оно заполняется автоматически по ФИО сотрудника, вошедшего в приложение. Также это поле ввода заблокировано от изменений – сотрудник может продавать товары только под своим именем.

325	Зрение	Maxima	55 comfort + ко...	941,0000	20
398	Зрение	OPTICLEAN	Opticlean conf...	12,0000	10
213	Зрение	Гельтек-Медика	Бл...		
014	Медицинские п...	OMRON	Тер...		
019	Гигиена	Kleenex	ант...		
050	Медицинские и	INFECTA	Ка...		

Оформление продажи

Штрихкод: 0723456789023

Количество: 2

Продал: Иванов И. И.

Оформить продажу

Описание пре

Подбор ана

Рис.2.2.6 Оформление продажи

Для оформления продажи реализована проверка на то, что количество товара, которое первостольник хочет продать, не превышает количество товара в наличии (рис.2.2.7). Для этого применяется скалярная функция checkStock (Приложение А, п.18).

Штрихкод	Категория	Бренд	Название	Цена	Мин.запас	В наличии
0023456789004	Медицинские п...	Armed	концентратор к...	26289,0000	1	21
0023456789031	Медицинские п...	AMRUS	Термометр мед...	1250,0000	20	30
0023456789071	Гигиена	Б.Браун Медик...	Эпиген интим 0...	1902,0000	5	35
0023456789076	Медицинские и...	SFM HOSPITAL	Шприц 50 мл 3-...	862,0000	50	84
0123456789009	Медицинские и...	DERMAGRIP	Перчатки смот...	1095,0000	30	88
0123456789036	Гигиена	Amra	диски ватные 1...	116,0000	30	84
0723456789023	Обезболивающ...	Реневал	Парацетамол	50,0000	40	260
0812345678958	Ортопедически...	КОМФОРТ	Корректор осе...	1250,0000	5	30
1000987654325	Зрение	Maxima	55 comfort + ко...	941,0000	20	3
1001234567898	Зрение	OPTICLEAN	Оп...			
1009876543213	Зрение	Гельтек-Медика	Бл...			
1123456789014	Медицинские п...	OMRON	Тер...			
1123456789019	Гигиена	Kleenex	ант...			
1123456789050	Медицинские и	INFECTA	Ка...			

Оформление продажи

Штрихкод: 1000987654325

Количество: 4

Продал: Иванов И. И.

Оформить продажу

Описание препаратов

Подбор аналогов

Рис.2.2.7 Предупреждение о количестве товара

Для продажи товаров, как и на других формах, где требуется ввести штрихкод товара, реализованы различные проверки корректности введенного штрихкода:

1. Ввод штрихкода длиной больше или меньше 13 цифр (рис.2.2.8).

Оформление продажи

Введен некорректный штрихкод

ОК

Оформление продажи

Штрихкод: 100098

Количество: 4

Продав: Иванов И. И.

Оформить продажу

Рис.2.2.8 Ошибка при вводе штрихкода (введено не 13 цифр)

2. Ввод каких-либо символов, кроме чисел (рис.2.2.9).

Оформление продажи

Введен некорректный штрихкод

ОК

Оформление продажи

Штрихкод: aaaaaaaaaaaaa

Количество: 4

Продав: Иванов И. И.

Оформить продажу

Рис.2.2.9 Ошибка при вводе штрихкода (введены не цифры)

3. Проверка существования введенного штрихкода в базе данных аптеки (рис.2.2.10), которая реализована с помощью скалярной функции checkBarcode (Приложение А, п.19).

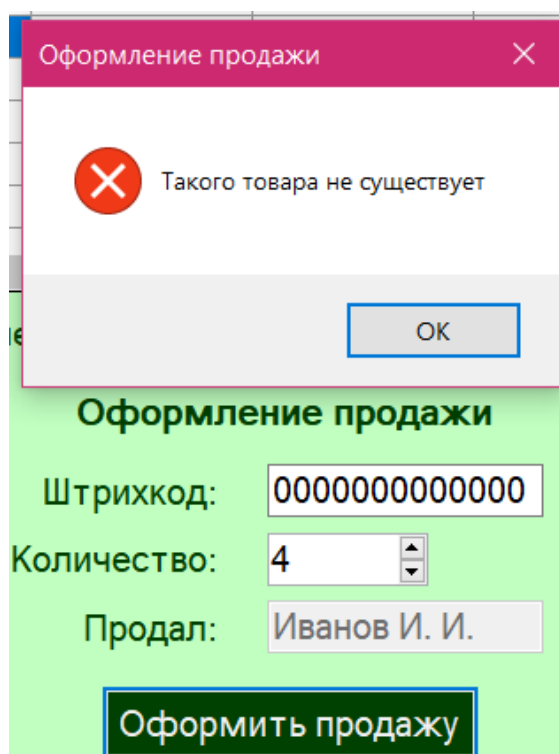


Рис.2.2.10 Ошибка при вводе штрихкода (несуществующий штрихкод)

Также первостольник может просматривать характеристики препаратов. При нажатии на кнопку «Описание препаратов» открывается соответствующая форма. Код формы представлен в приложении В, п.3. Изначально форма пустая, требуется ввести штрихкод препарата, подробные характеристики которого требуется посмотреть, и нажать кнопку «смотреть» (рис.2.2.11). При нажатии на кнопку срабатывает табличная функция «seeDescriptionOfMedicament» (Приложение А, п.4). В поле «Искомый товар» выводятся бренд, название и цена товара, для которого производится поиск. Для вывода данной строки применяется скалярная функция «getProductInfo» (Приложение А, п.5).

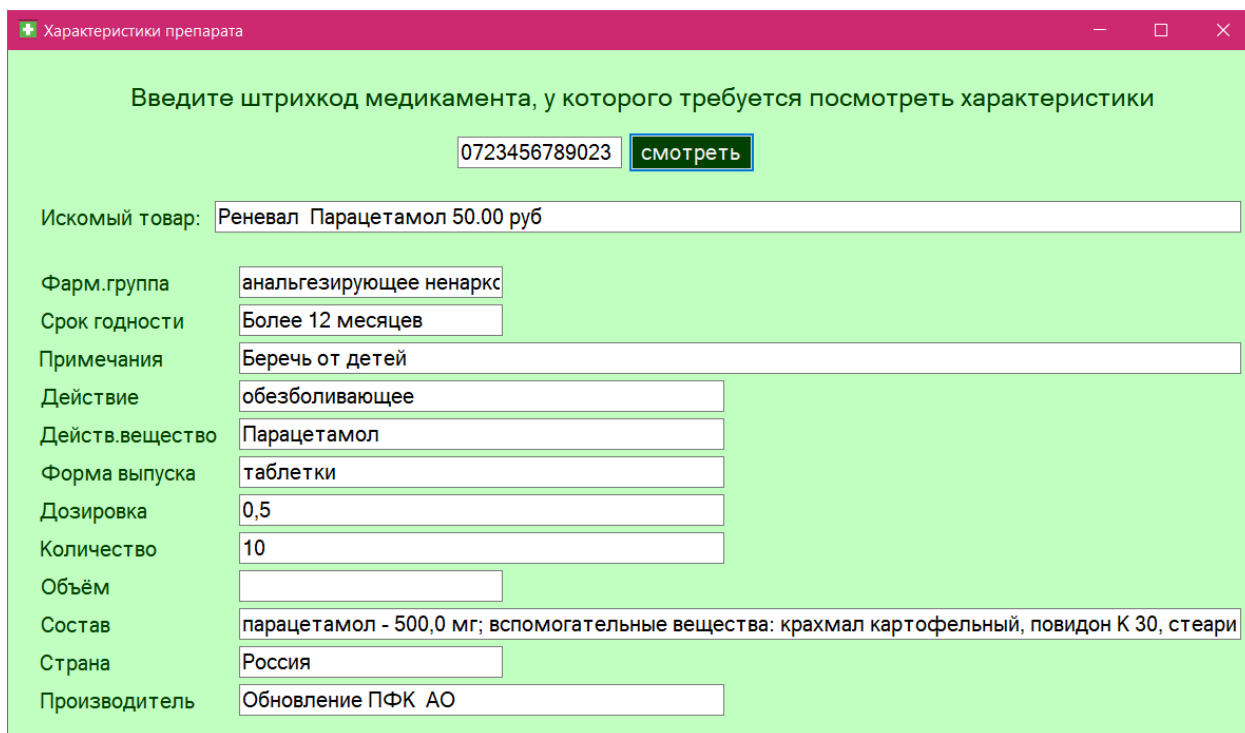


Рис.2.2.11 Просмотр характеристик медикамента

Для поля ввода штрихкода реализованы проверки корректности, представленные на рис.2.2.8-2.2.10. Кроме того, при вводе штрихкода товара, для которого отсутствуют характеристики, выводится соответствующее сообщение (рис.2.2.12).

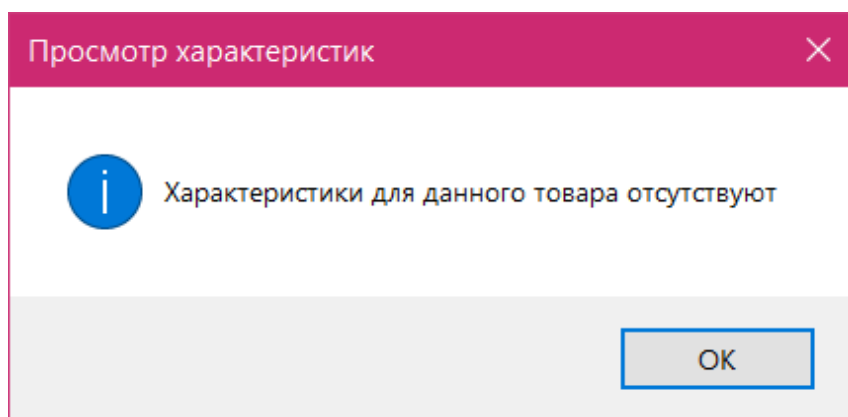


Рис.2.2.12 Сообщение об отсутствии характеристик для товара

Еще первостольник может производить поиск аналогов товаров. При нажатии на кнопку «Подбор аналогов» открывается соответствующая форма. Код формы представлен в приложении В, п.4. Изначально форма пустая, требуется ввести штрихкод товара, к которому требуется подобрать аналоги, и нажать кнопку «найти» (рис.2.2.13). При нажатии на кнопку срабатывает табличная функция «findAnalog» (Приложение А, п.6). Для поля ввода

штрихкода реализованы проверки корректности, представленные на рис.2.2.8-2.2.10. В поле «Искомый товар» выводятся бренд, название и цена товара для которого производится поиск. Для вывода данной строки применяется скалярная функция «getProductInfo» (Приложение А, п.5).

Введите штрихкод товара, для которого требуется найти аналоги

9923456789099

Искомый товар:

Сурах менструальная чаша силиконовая размер regular/розовая 410.00 руб

	Штрихкод	Бренд	Название	Цена	В наличии
▶	6623456789037	Onlycup	менструальная чаша серия лен размер...	554,0000	60
	9923456781462	DULI	менструальная чаша	400,0000	60

Если ничего не отображается, это значит, что аналоги отсутствуют

Рис.2.2.13 Подбор аналогов

На форме первостольника, как и на многих других формах в данном приложении, есть кнопка «Назад», которая при нажатии возвращает на форму, с которой пользователь попал на текущую. В данном случае первостольник со своей формы вернется на форму авторизации. Эта кнопка требуется для таких ситуаций, как возвращение на предыдущую форму при выполнении нужных действий на текущей или когда у сотрудника заканчивается рабочая смена, и ему нужно выйти на форму авторизации, чтобы сотрудник, пришедший на следующую смену, мог авторизоваться под своим именем.

На этом функции на форме первостольника заканчиваются, перейдем к функциям менеджера. При входе в аккаунт менеджера, у пользователя отображается форма, представленная на рисунке 2.2.14. Код формы представлен в приложении В, п.5. Менеджеру доступен переход на форму учета товаров или учета сотрудников при нажатии соответствующих кнопок.

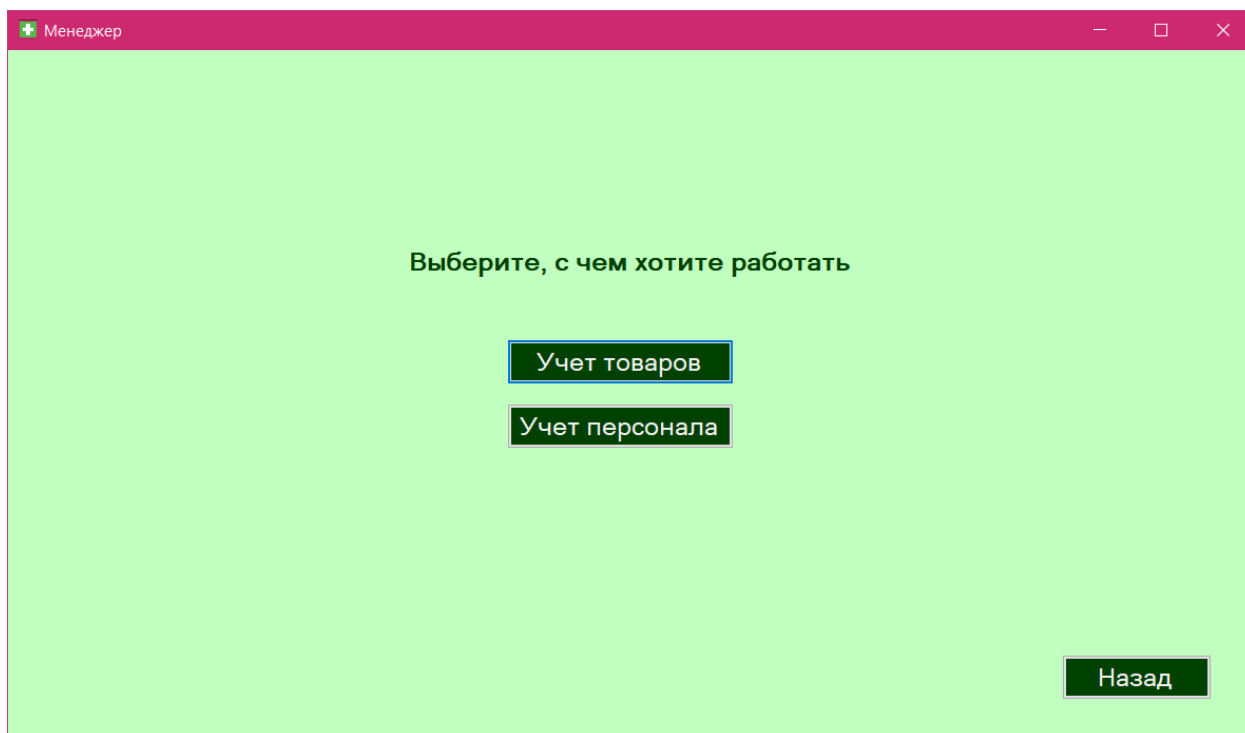


Рис.2.2.14 Форма менеджера

Начнем с кнопки «Учет товаров» (рис.2.2.15). При нажатии на неё отображается форма «Товары», программный код которой представлен в приложении В, п.6.

Штрихкод	Категория	Бренд	Названия	Цена	Мин.запас	В наличии
0023456789004	Медицинские п...	Armed	концентратор к...	26289,0000	1	21
0023456789031	Медицинские п...	AMRUS	Термометр мед...	1250,0000	20	30
0023456789071	Гигиена	Б. Браун Медик...	Эпиген интим 0...	1902,0000	5	35
0023456789076	Медицинские и...	SFM HOSPITAL	Шприц 50 мл 3-...	862,0000	50	84
0123456789009	Медицинские и...	DERMAGRIP	Перчатки смот...	1095,0000	30	88
0123456789036	Гигиена	Amra	диски ватные 1...	116,0000	30	84
0723456789023	Обезболивающ...	Реневал	Парацетамол	50,0000	40	260
0812345678958	Ортопедически...	КОМФОРТ	Корректор оса...	1250,0000	5	30
1000987654325	Зрение	Maxima	55 comfort + ко...	941,0000	20	3
1001234567898	Зрение	OPTICLEAN	Opticlean салф...	42,0000	10	14
1009876543213	Зрение	Гельтек-Медика	Блефаросалфе...	505,0000	5	29
1123456789014	Медицинские п...	OMRON	Термометр omr...	563,0000	30	109
1123456789019	Гигиена	Kleenex	антибактериал...	174,0000	20	58

Рис.2.2.15 Форма учета товаров

Здесь менеджер может просматривать таблицу с товарами, где указаны их цены и количество в наличии. Таблица товаров выводится с помощью

представления «tableProduct» (Приложение А, п.2). Менеджер может производить поиск по штрихкоду, бренду, названию или категории товара: в поле вводятся необходимые данные, при нажатии на кнопку «Найти» осуществляется поиск по всем возможным совпадениям (рис.2.2.16). Вернуть изначальный вид таблицы можно с помощью кнопки «Сбросить», при ее нажатии также очищается поле ввода.

Штрихкод	Категория	Бренд	Названия	Цена	Мин.запас	В наличии
0123456789036	Гигиена	Amga	диски ватные 1...	116,0000	30	84

Введите штрихкод, бренд, название или категорию товара:

Рис.2.2.16 Поиск по товарам

Менеджер может оформлять поставки товаров. Для этого требуется заполнить поля «Штрихкод», «Количество», выбрать поставщика из выпадающего списка и нажать кнопку «Оформить поставку» (рис.2.2.17). Выпадающий список поставщиков реализован с помощью табличной функции «getPurveyor» (приложение А, п.7). При нажатии на кнопку срабатывает хранимая процедура «addPurchase» (приложение А, п.8). Поле сотрудника, заказывающего товар, заполнять не требуется, так как оно заполняется автоматически по ФИО сотрудника, вошедшего в приложение. Также это поле ввода заблокировано от изменений – сотрудник может заказывать товары только под своим именем. Для поля ввода штрихкода реализованы проверки корректности, представленные на рис.2.2.8-2.2.10.

Рис.2.2.17 Оформление поставки

На рисунке 2.2.18 можно заметить, что количество товара, для которого на рисунке 2.2.17 была оформлена поставка, увеличилось на 50 единиц по сравнению с количеством этого товара на рисунке 2.2.16.

Штрихкод	Категория	Бренд	Названия	Цена	Мин.запас	В наличии
0123456789036	Гигиена	Amga	диски ватные 1...	116,0000	30	134

ите штрихкод, бренд, название или категорию товара:

0123456789036

Найти

Сбросить

Рис.2.2.18 Проверка обновления количества товара после поставки

В случае, если менеджеру требуется произвести заказ у нового поставщика, он должен добавить нового поставщика, нажав на кнопку «Добавить поставщика». При нажатии на кнопку открывается соответствующая форма (рис.2.2.19), программный код которой представлен в приложении В, п.7.

	Название	Представитель	Страна	Город	Адрес	Телефон	Оплата
▶	Аделейд	Казачков Б.Л.	Беларусь	Минск	ул.Стрелецкая, 9	788369	безнал
	ПрофитМед	Сидоренко А.В.	Россия	Москва	ул.Мира, 1	742330	
	НПО Гарант	Кошка А.Е.	Россия	Рязань	ул.Дружбы, 3	541205	безнал
	МедТорг	Румянцев И.И.	Россия	Белгород	ул.Энгельса, 16	741568	наличные
	ООО МедФарм	Зорьин А.Г.	Россия	Химки	ул.Часовая, 9	850023	безнал
	ООО Север	Овчаров Д.А.	Украина	Киев	ул.Ленина, 46	852001	корр.счет 4660
	ИП Москвич	Рыбкин П.Б.	Россия	Москва	ул.Южная, 10	856010	

Проверьте наличие поставщика по его названию: **Найти** **Сбросить**

Добавление поставщика

Название

Представитель

Страна

Город

Адрес

Телефон

Способ оплаты

Добавить **Назад**

Рис.2.2.19 Форма добавления поставщиков

На данной форме есть таблица с поставщиками, где можно посмотреть все данные о поставщиках, информация о которых есть в базе данных аптеки. Можно осуществлять поиск по названию поставщика, введя его в соответствующее поле и нажав кнопку «Найти» (рис.2.2.20). Вернуть изначальный вид таблицы можно с помощью кнопки «Сбросить», при ее нажатии также очищается поле ввода.

	Название	Представитель	Страна	Город	Адрес	Телефон	Оплата
▶	Аделейд	Казачков Б.Л.	Беларусь	Минск	ул.Стрелецкая, 9	788369	безнал

Проверьте наличие поставщика по его названию: **Найти** **Сбросить**

Рис.2.2.20 Поиск по названию поставщика

Основная функция данной формы – добавление новых поставщиков. Для этого требуется заполнить поля «Название», «Представитель», «Страна» (по умолчанию – «Россия»), «Город», «Адрес», «Телефон», «Способ оплаты» и нажать кнопку «Добавить» (рис.2.2.21). При нажатии на кнопку срабатывает хранимая процедура «addPurveyor» (приложение А, п.9). Также реализована

проверка на заполнение всех полей, кроме способа оплаты, – если хоть одно будет пустым, выведется ошибка (рис.2.2.22).

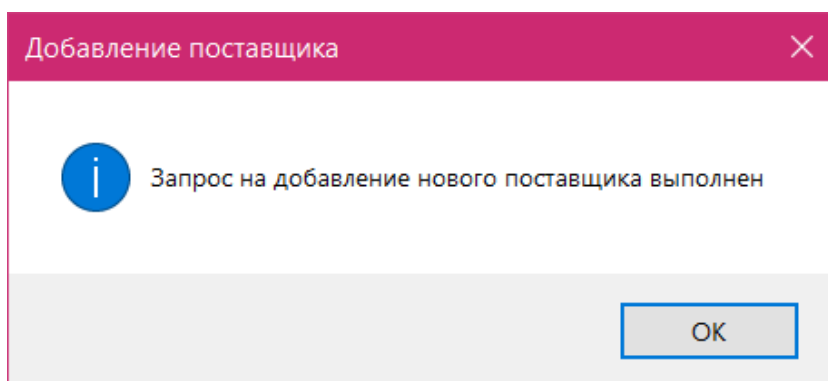


Рис.2.2.21 Добавление поставщика

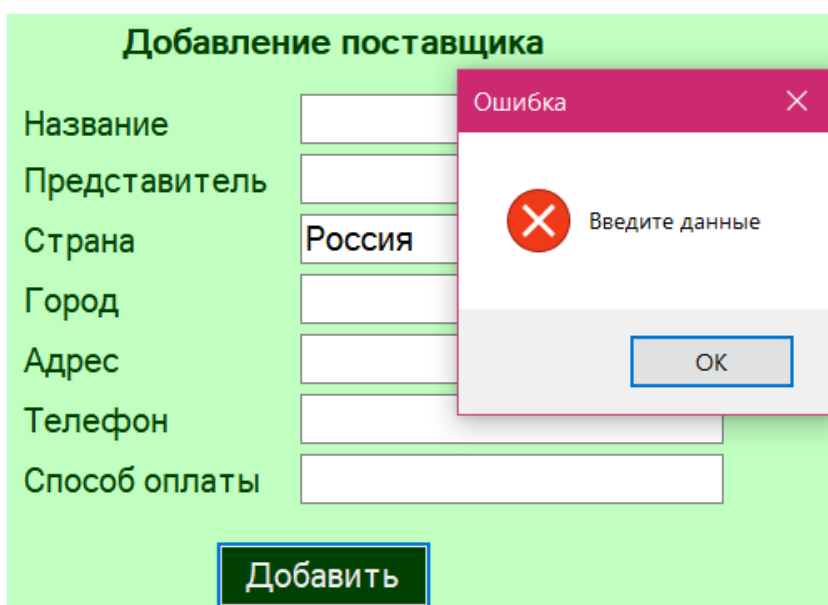
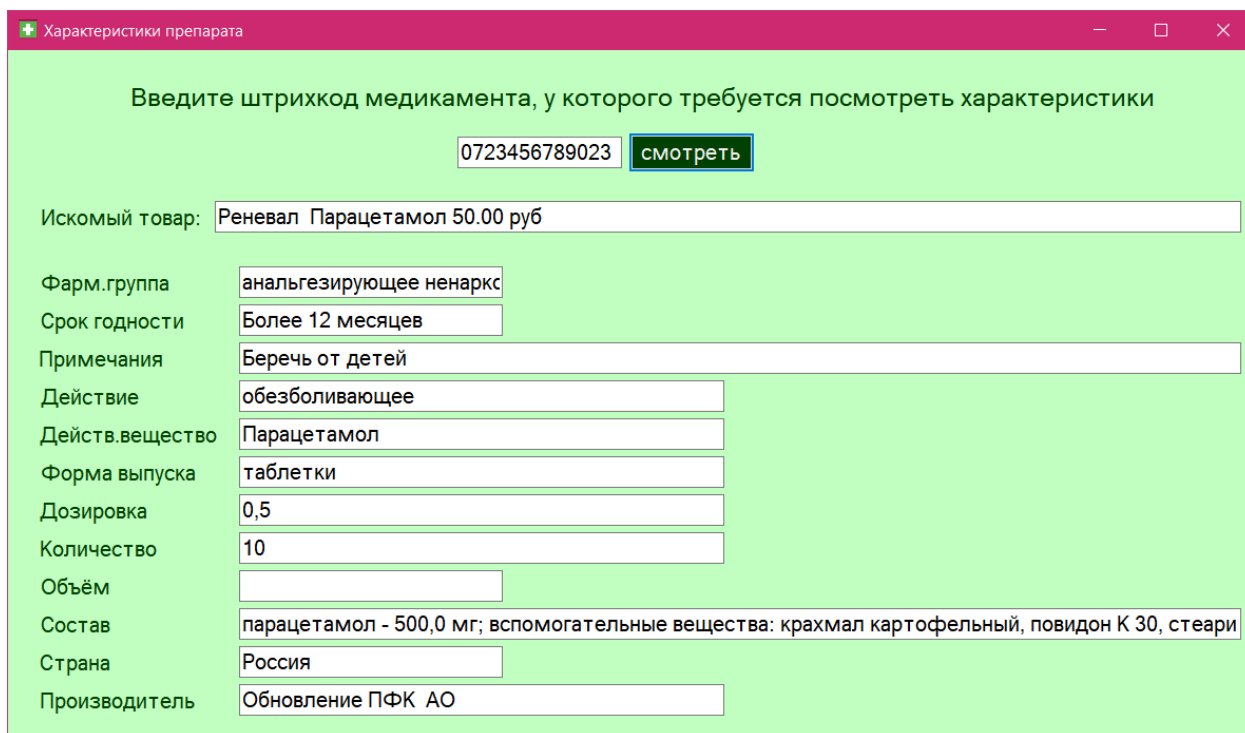


Рис.2.2.22 Проверка заполнения полей

Менеджер, как и первостольник, может просматривать характеристики препаратов. При нажатии на кнопку «Описание препаратов» открывается соответствующая форма. Код формы представлен в приложении В, п.3. Изначально форма пустая, требуется ввести штрихкод препарата, подробные характеристики которого требуется посмотреть, и нажать кнопку «смотреть» (рис.2.2.23). При нажатии на кнопку срабатывает табличная функция «seeDescriptionOfMedicament» (Приложение А, п.4). В поле «Искомый товар» выводятся бренд, название и цена товара, для которого производится поиск. Для вывода данной строки применяется скалярная функция «getProductInfo» (Приложение А, п.5).



Характеристики препарата

Введите штрихкод медикамента, у которого требуется посмотреть характеристики

0723456789023

Искомый товар: Реневал Парацетамол 50.00 руб

Фарм.группа:

Срок годности:

Примечания:

Действие:

Действ.вещество:

Форма выпуска:

Дозировка:

Количество:

Объём:

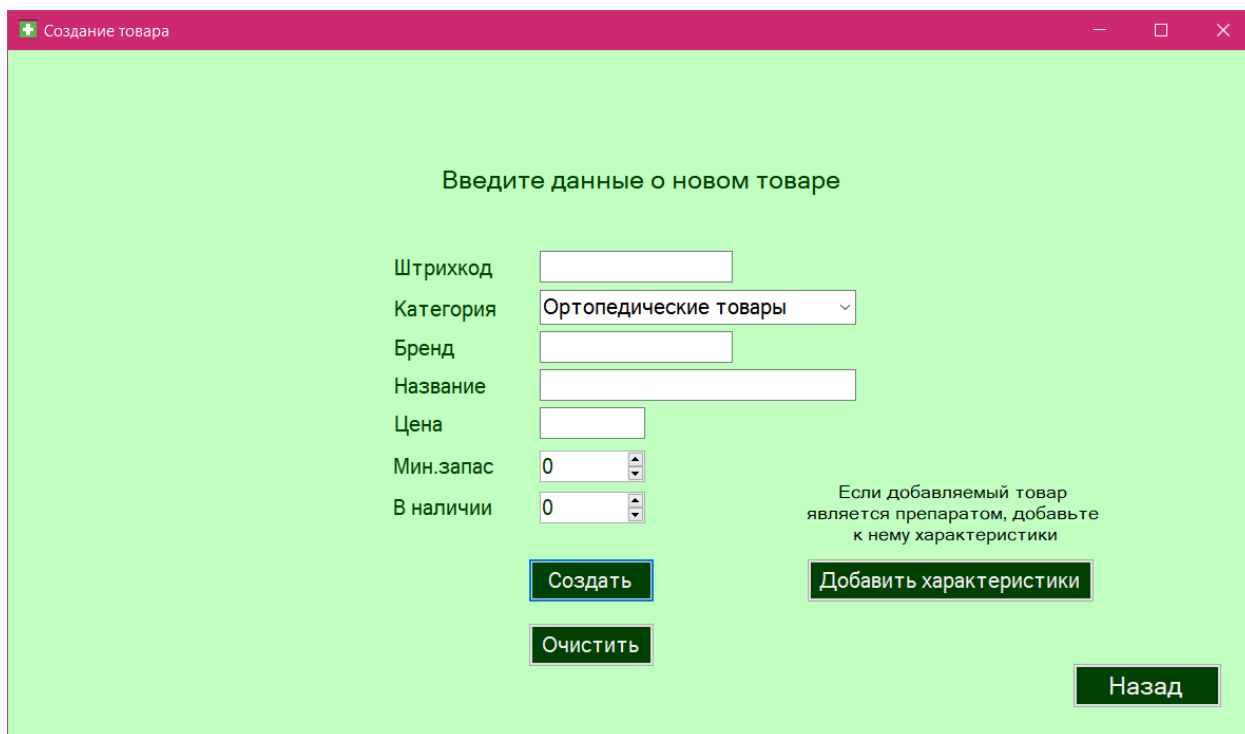
Состав:

Страна:

Производитель:

Рис.2.2.23 Просмотр характеристик медикамента

По кнопке «Добавить товар» менеджер открывает форму «Создание товара» (рис.2.2.24), программный код которой представлен в приложении Б, п.8.



Создание товара

Введите данные о новом товаре

Штрихкод:

Категория:

Бренд:

Название:

Цена:

Мин.запас:

В наличии:

Если добавляемый товар является препаратом, добавьте к нему характеристики

Рис.2.2.24 Форма создания товара

Для создания нового товара требуется заполнить поля «Штрихкод», «Категория» из выпадающего списка, «Бренд», «Название», «Цена»,

«Минимальный запас», «В наличии» и нажать кнопку «Создать» (рис.2.2.25). При нажатии на кнопку срабатывает хранимая процедура «addProduct» (приложение А, п.10). Выпадающий список категорий товаров берётся из таблицы Category по значениям атрибута «NameCategory». Также реализована проверка на заполнение всех полей – если хоть одно будет пустым, выведется ошибка (рис.2.2.26). Для поля ввода штрихкода реализованы проверки корректности, представленные на рис.2.2.8-2.2.10. Также в данной форме имеется проверка на корректность вводимой цены – если будет введено что-то кроме цифр, то выведется ошибка (рис.2.2.27).

The screenshot shows a form titled "Введите данные о новом товаре" (Enter data for a new product). The form fields are filled with the following data: Штрихкод (Barcode) is 9874123651239; Категория (Category) is "Витамины и добавки" (Vitamins and supplements); Бренд (Brand) is "Биокор" (Biocor); Название (Name) is "Пустырник 50 шт" (Motherwort 50 pieces); Цена (Price) is 50; Мин.запас (Min. stock) is 0; В наличии (In stock) is 0. A green "Создать" (Create) button is at the bottom. A modal dialog titled "Создание товара" (Product creation) is open, showing a blue information icon and the text "Товар создан" (Product created), with an "ОК" button.

Рис.2.2.25 Создание товара

The screenshot shows the same form as in the previous image, but with some fields empty: Штрихкод is 1234569512365; Категория is "Витамины и добавки"; Бренд, Название, and Цена are empty. The "Мин.запас" and "В наличии" fields are 0. A green "Создать" button is at the bottom. A modal dialog titled "Ошибка" (Error) is open, showing a red error icon and the text "Введены не все требуемые данные" (Not all required data is entered), with an "ОК" button.

Рис.2.2.26 Проверка заполнения полей

Введите данные о новом товаре

Штрихкод 1234569512365

Категория Витамины и добавки

Бренд Биокор

Название Валерьянка

Цена 0000

Мин.запас 0

В наличии 0

Создать

Добавление товара

Введена некорректная цена

OK

Рис.2.2.27 Проверка ввода цены

На форме есть кнопка «Добавить характеристики», которую требуется нажать в ситуации, если добавляемый товар является препаратом. При нажатии на кнопку открывается форма добавления характеристик (рис.2.2.28), программный код которой представлен в приложении В, п.9. Для создания характеристик к препарату требуется заполнить поля «Штрихкод», «Фарм.группа», «Срок годности» (по умолчанию «Более 12 месяцев»), «Примечания» (по умолчанию «Беречь от детей»), «Действие», «Действующее вещество», «Форма выпуска», «Дозировка», «Количество», «Объем», «Состав», «Страна», «Производитель» и нажать кнопку «Добавить» (рис.2.2.29). При нажатии на кнопку срабатывает хранимая процедура «addMedicament» (приложение А, п.11). Также реализована проверка на заполнение полей «Штрихкод», «Фарм.группа», «Срок годности», «Действующее вещество», «Форма выпуска», «Состав», «Производитель» – если хоть одно из них будет пустым, выведется ошибка (рис.2.2.30). Для поля ввода штрихкода реализованы проверки корректности, представленные на рис.2.2.8-2.2.10. Также в данной форме имеется проверка на уникальность вводимого штрихкода в таблице описаний медикаментов – если

характеристики такого товара уже существуют, то выведется ошибка (рис.2.2.31). На форме есть кнопка «Очистить», которая очищает все поля, кроме тех, где заданы значения по умолчанию.

The screenshot shows a web application window titled "Добление характеристик" (Adding characteristics). The main heading is "Введите данные о препарате" (Enter data about the drug). The form contains the following fields:

- Штрихкод (Barcode): empty text input
- Фарм.группа (Pharm. group): empty text input
- Срок годности (Shelf life): dropdown menu with "Более 12 месяцев" (More than 12 months) selected
- Примечания (Remarks): text input with "Беречь от детей" (Keep away from children)
- Действие (Action): empty text input
- Действ.вещество (Active substance): empty text input
- Форма выпуска (Form of release): empty text input
- Дозировка (Dosage): empty text input
- Количество (Quantity): spinner box with "0"
- Объём (Volume): empty text input
- Состав (Composition): empty text input
- Страна (Country): empty text input
- Производитель (Manufacturer): empty text input

At the bottom right, there are two buttons: "Добавить" (Add) and "Очистить" (Clear).

Рис.2.2.28 Форма добавления характеристик препаратов

This screenshot shows the same form as Figure 2.2.28, but with data entered and a confirmation dialog box open.

Form Data:

- Штрихкод: 9874123651239
- Фарм.группа: БАД
- Срок годности: Более 12 месяцев
- Примечания: Беречь от детей
- Действие: (empty)
- Действ.вещество: Пустырник
- Форма выпуска: драже
- Дозировка: 0,18
- Количество: 50
- Объём: (empty)
- Состав: медицинское, тиамина гидрохлорид (витамин B1), пиридоксина гидрохлорид (витамин B6)
- Страна: Россия
- Производитель: Биокор Фирма ООО

Confirmation Dialog: A small window titled "Создание описания товара" (Creating item description) is displayed over the "Действ.вещество" field. It contains an information icon and the text "Характеристики к товару добавлены" (Characteristics added to the item), with an "OK" button.

The "Добавить" and "Очистить" buttons are still visible at the bottom right.

Рис.2.2.29 Добавление характеристик

Введите данные о препарате

Штрихкод	<input type="text" value="0123456789036"/>
Фарм.группа	<input type="text"/>
Срок годности	<input type="text" value="Более 12 месяцев"/>
Примечания	<input type="text" value="Беречь от детей"/>
Действие	<input type="text"/>
Действ.вещество	<input type="text"/>
Форма выпуска	<input type="text"/>
Дозировка	<input type="text" value="0,18"/>
Количество	<input type="text" value="50"/>
Объём	<input type="text"/>
Состав	<input type="text"/>
Страна	<input type="text"/>
Производитель	<input type="text"/>

Добавить **Очи**

Ошибка

✖

Введены не все требуемые данные (проверьте заполнение следующих полей: Штрихкод, Фарм.группа, Срок годности, Действующее вещество, Форма выпуска, Состав, Производитель)

ОК

Рис.2.2.30 Проверка ввода данных

Введите данные о препарате

Штрихкод	<input type="text" value="9874123651239"/>
Фарм.группа	<input type="text"/>
Срок годности	<input type="text" value="Более 12 месяцев"/>
Примечания	<input type="text" value="Беречь от детей"/>
Действие	<input type="text"/>

Ошибка

✖

Описание этого товара уже имеется.

ОК

Рис.2.2.31 Проверка существования характеристик

На основной форме учета товаров есть кнопка «Списать товар», которая направляет на форму списания товаров (рис.2.2.32), программный код которой представлен в приложении В, п.10. Для списания товара требуется заполнить поля «Штрихкод», «Количество», «Причина списания» (по умолчанию «Истекает срок годности») и нажать кнопку «Списать» (рис.2.2.33). При нажатии на кнопку срабатывает хранимая процедура «writeOffProduct» (приложение А, п.12). Для поля ввода штрихкода реализованы проверки корректности, представленные на рис.2.2.8-2.2.10. В поле «Искомый товар» выводятся бренд, название и цена товара, для которого производится поиск.

Для вывода данной строки применяется скалярная функция «getProductInfo» (Приложение А, п.5).

Рис.2.2.32 Форма списания товаров

Рис.2.2.33 Списание товара

На форме списания товаров есть кнопка «Списанные товары», которая направляет на форму списанных товаров (рис.2.2.34). Код формы представлен в приложении В п.11. Таблица списанных товаров выводится с помощью представления «deleteProductForWF» (Приложение А, п.13).

Списанные товары

Информация о списаниях товаров

	№списания	Штрихкод	Бренд	Название	Количество	Дата	Причина
▶	1	6323456789093	MOLDEX	Беруши moldex ...	2	01.09.2022	Истекает срок ...
	2	0023456789031	AMRUS	Термометр мед...	5	10.12.2023	Истекает срок ...
	3	0023456789071	Б.Браун Медик...	Эпиген интим 0...	10	10.12.2023	Истекает срок ...
	4	0023456789071	Б.Браун Медик...	Эпиген интим 0...	2	10.12.2023	Истекает срок ...
	5	0023456789071	Б.Браун Медик...	Эпиген интим 0...	3	10.12.2023	Истекает срок ...
	6	0123456789036	Amra	диски ватные 1...	16	19.12.2023	Товар испорчен
	7	1000987654325	Maxima	55 comfort + ко...	6	19.12.2023	Истекает срок ...
	8	9923456789058	AMRUS	Термометр мед...	6	19.12.2023	Истекает срок ...
	9	1123456789014	OMRON	Термометр омг...	6	19.12.2023	Истекает срок ...
	10	0023456789076	SFM HOSPITAL	Шприц 50 мл 3-...	16	19.12.2023	Товар испорчен
	11	0723456789023	Реневал	Парацетамол	8	19.12.2023	Истекает срок ...
	12	0723456789023	Реневал	Парацетамол	3	19.12.2023	Истекает срок ...
	13	1001234567898	OPTICLEAN	Opticlean салф...	3	19.12.2023	Нарушены усло...
	14	0812345678958	КОМФОРТ	Корректор оса...	5	20.12.2023	Истекает срок ...
	15	1000987654325	Maxima	55 comfort + ко...	15	20.12.2023	Истекает срок ...
	16	1000987654325	Maxima	55 comfort + ко...	1	20.12.2023	Истекает срок ...
	17	0123456789036	Amra	диски ватные 1...	3	26.12.2023	Истекает срок ...

Введите штрихкод, бренд или название товара:

Рис.2.2.34 Форма просмотра списанных товаров

На данной форме можно осуществить поиск по штрихкоду, бренду или названию товара (рис.2.2.35). Вернуть изначальный вид таблицы можно с помощью кнопки «Сбросить», при ее нажатии также очищается поле ввода.

Информация о списаниях товаров

	№списания	Штрихкод	Бренд	Название	Количество	Дата	Причина
▶	11	0723456789023	Реневал	Парацетамол	8	19.12.2023	Истекает срок ...
	12	0723456789023	Реневал	Парацетамол	3	19.12.2023	Истекает срок ...
	18	0723456789023	Реневал	Парацетамол	10	26.12.2023	Истекает срок ...

Введите штрихкод, бренд или название товара:

Рис.2.2.35 Поиск по списанным товарам

На основной форме учета товаров также есть кнопка «Списанные товары», которая направляет на ту же форму просмотра списанных товаров, что и кнопка на форме списания товаров.

Также на основной форме учета товаров есть кнопка «Продажи», которая направляет на форму просмотра продаж (рис.2.2.36), программный код которой представлен в приложении В, п.12. Таблица проданных товаров выводится с помощью представления «viewSell» (Приложение А, п.14). На форме можно осуществить поиск по штрихкоду, бренду или названию товара (рис.2.2.37). Вернуть изначальный вид таблицы можно с помощью кнопки «Сбросить», при ее нажатии также очищается поле ввода.

Информация о продажах товаров								
	№чека	ФИО	Дата	Штрихкод	Бренд	Название	Количество	Суммарная стоимость
▶	2	Аксенов Л. В.	03.11.2022	1123456789019	Kleenex	антибактериал...	2	348,0000
	2	Аксенов Л. В.	03.11.2022	6323456789093	MOLDEX	Беруши moldex ...	2	550,0000
	3	Лосев В. М.	12.12.2022	0023456789004	Armed	концентратор к...	2	52578,0000
	4	Корнева В. М.	29.01.2023	1009876543213	Гельтек-Медика	Блефаросалфе...	1	505,0000
	5	Корнева В. М.	15.02.2023	1123456789014	OMRON	Термометр omr...	1	563,0000
	6	Аксенов Л. В.	30.03.2023	0723456789023	Реневал	Парацетамол	2	100,0000
	7	Аксенов Л. В.	08.04.2023	0023456789076	SFM HOSPITAL	Шприц 50 мл 3-...	1	862,0000
	8	Орлов Е. М.	16.05.2023	0023456789004	Armed	концентратор к...	1	26289,0000
	9	Аксенов Л. В.	01.06.2023	2001234567897	Avizor	aqua soft раств...	1	400,0000
	10	Иванов И. И.	12.12.2023	1123456789014	OMRON	Термометр omr...	2	1126,0000
	11	Иванов И. И.	12.12.2023	6323456789093	MOLDEX	Беруши moldex ...	2	550,0000
	12	Иванов И. И.	12.12.2023	1009876543213	Гельтек-Медика	Блефаросалфе...	2	1010,0000
	13	Иванов И. И.	12.12.2023	0123456789009	DERMAGRIP	Перчатки смот...	2	2190,0000
	14	Иванов И. И.	12.12.2023	9923456789012	ЛИНИЯ ЗДОР...	Аппарат магни...	1	7267,0000
	15	Иванов И. И.	12.12.2023	8612345678909	Лизобакт комп...	0,0001+0,004+0...	3	1200,0000
	16	Иванов И. И.	12.12.2023	9923456789012	ЛИНИЯ ЗДОР...	Аппарат магни...	2	14534,0000
	17	Иванов И. И.	12.12.2023	9923456789012	ЛИНИЯ ЗДОР...	Аппарат магни...	2	14534,0000
	18	Иванов И. И.	20.12.2023	1823456789007	МПК Елец	Шприц 50 мл тр...	3	3033,0000

Введите штрихкод, бренд или название товара:

Рис.2.2.36 Информация о продажах товаров

Проданные товары

Информация о продажах товаров

№чека	ФИО	Дата	Штрихкод	Бренд	Название	Количество	Суммарная стоимость
6	Аксенов Л. В.	30.03.2023	0723456789023	Реневал	Парацетамол	2	100,0000
28	Корнева В. М.	20.12.2023	0723456789023	Реневал	Парацетамол	2	100,0000
29	Корнева В. М.	20.12.2023	0723456789023	Реневал	Парацетамол	5	250,0000
40	Иванов И. И.	26.12.2023	0723456789023	Реневал	Парацетамол	2	100,0000

Введите штрихкод, бренд или название товара:

Рис.2.2.37 Поиск по проданным товарам

Также на основной форме учета товаров есть кнопка «Поставки», которая направляет на форму просмотра поставок (рис.2.2.38), программный код которой представлен в приложении В, п.13. Таблица закупленных товаров выводится с помощью представления «viewPurchase» (Приложение А, п.15). На форме можно осуществить поиск по штрихкоду, бренду или названию товара (рис.2.2.39). Вернуть изначальный вид таблицы можно с помощью кнопки «Сбросить», при ее нажатии также очищается поле ввода.

Закупленные товары

Информация о закупленных товарах

Поставщик	Заказчик	Дата	Штрихкод	Бренд	Название	Цена	Количество
Аделейд	Лаврова Д. А.	30.08.2022	6323456789093	MOLDEX	Беруши moldex ...	150,0000	150
ПрофитМед	Горюнов Г. Г.	12.10.2022	0023456789004	Armed	концентратор к...	410,0000	10
НПО Гарант	Горюнов Г. Г.	31.12.2022	1596324788741	Lactacyd	салфетки для и...	410,0000	100
МедТорг	Горюнов Г. Г.	30.01.2023	2023202320232	Артека	стельки	410,0000	40
ООО МедФарм	Лаврова Д. А.	10.02.2023	2023202320232	Артека	стельки	410,0000	40
ООО Север	Горюнов Г. Г.	25.02.2023	9923456789012	ЛИНИЯ ЗДОР...	Аппарат магни...	410,0000	15
ИП Медичек	Лаврова Д. А.	05.04.2023	5901234567890	ИСТОК-АУДИО...	Аппарат слухов...	410,0000	5
ИП Медичек	Лаврова Д. А.	05.11.2023	3012345678914	B WELL	Корсет ortho w...	410,0000	15
Orion Pharma	Лаврова Д. А.	20.12.2023	3012345678914	B WELL	Корсет ortho w...	410,0000	15
Orion Pharma	Горюнов Г. Г.	20.12.2023	3012345678914	B WELL	Корсет ortho w...	410,0000	15
Orion Pharma	Горюнов Г. Г.	20.12.2023	1596324788741	Lactacyd	салфетки для и...	410,0000	50
Orion Pharma	Горюнов Г. Г.	20.12.2023	1001234567898	OPTICLEAN	Opticlean салф...	410,0000	15
Аделейд	Горюнов Г. Г.	20.12.2023	0723456789023	Реневал	Парацетамол	410,0000	40
Аделейд	Горюнов Г. Г.	20.12.2023	0723456789023	Реневал	Парацетамол	410,0000	40
Аделейд	Горюнов Г. Г.	20.12.2023	0023456789071	Б.Браун Медик...	Эпиген интим О...	410,0000	20
Аделейд	Горюнов Г. Г.	20.12.2023	1234567891124	Коделак	Сироп от кашл...	410,0000	40
Аделейд	Горюнов Г. Г.	20.12.2023	2223456789033	ТЕРМО-КОНТ ...	Термоконтейне...	410,0000	40

Вы можете осуществить поиск по поставщику, заказчику, штрихкоду, бренду или названию товара

Рис.2.2.38 Информация о закупленных товарах

Закупленные товары

Информация о закупленных товарах

	Поставщик	Заказчик	Дата	Штрихкод	Бренд	Название	Цена	Количество
	ИП Медичек	Лаврова Д. А.	05.11.2023	3012345678914	B WELL	Корсет ortho w...	410,0000	15
	Orion Pharma	Лаврова Д. А.	20.12.2023	3012345678914	B WELL	Корсет ortho w...	410,0000	15
▶	Orion Pharma	Горюнов Г. Г.	20.12.2023	3012345678914	B WELL	Корсет ortho w...	410,0000	15

3012345678914

Вы можете осуществить поиск по поставщику, заказчику, штрихкоду, бренду или названию товара

Рис.2.2.39 Поиск по закупленным товарам

На главной форме менеджера есть кнопка «Учет сотрудников», которая направляет на соответствующую форму (рис.2.2.40). Код формы представлен в приложении В, п.

Сотрудники

	Идентификатор	ФИО	Должность	Email	Телефон
▶	1	Орлов Е. М.	Провизор	vanyaivanovv@...	760522
	2	Лаврова Д. А.	Менеджер	petrov@gmail.c...	654951
	3	Тимофеева П. М.	Фармацевт	kolyaaa@yande...	332226
	4	Горюнов Г. Г.	Менеджер	grgrgt@mail.ru	999999
	5	Аксенов Л. В.	Провизор	alexalex@gmail....	574666
	6	Лосев В. М.	Уволен Уборщик		852304
	7	Корнева В. М.	Провизор	eee@mail.ru	741560
	8	Никитин Н. Н.	Директор	nikime@gmail.c...	333333
	9	Алексеев С. Д.	Уволен Фарма...	alexaisd@yahoo....	369149
	10	Иванов И. И.	Провизор	iii@mail.com	123456
	11	Кошкин Н. Д.	Уборщик		123654

Добавление сотрудника

ФИО

Должность

Почта

Телефон

Выберите сотрудника в таблице

Рис.2.2.40 Форма учета сотрудников

На данной форме есть функция добавления сотрудника. Для добавления нового сотрудника требуется заполнить поля «ФИО», «Должность» из

выпадающего списка, «Почта», «Телефон» и нажать кнопку «Добавить сотрудника» (рис.2.2.41). При нажатии на кнопку срабатывает хранящая процедура «addSotr» (приложение А, п.16). Выпадающий список категорий товаров задан в свойствах формы и содержит в себе 5 должностей – Провизор, Фармацевт, Менеджер, Уборщик, Директор. Также реализована проверка на заполнение полей «Фино» и «Должность» – если хоть одно из них будет пустым, выведется ошибка (рис.2.2.42). Также в данной форме имеется проверка уникальности вводимого номера телефона – если в базе данных уже имеется сотрудник с таким номером, то выведется ошибка (рис.2.2.43).

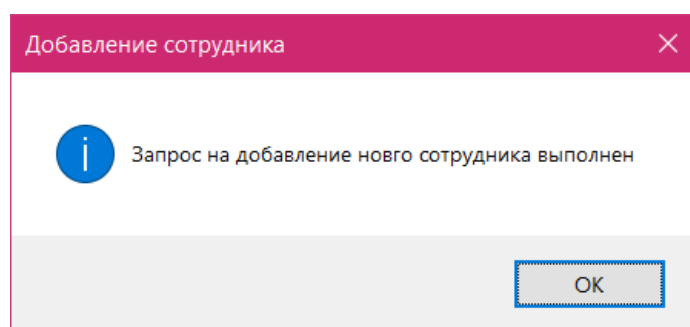


Рис.2.2.41 Добавление нового сотрудника

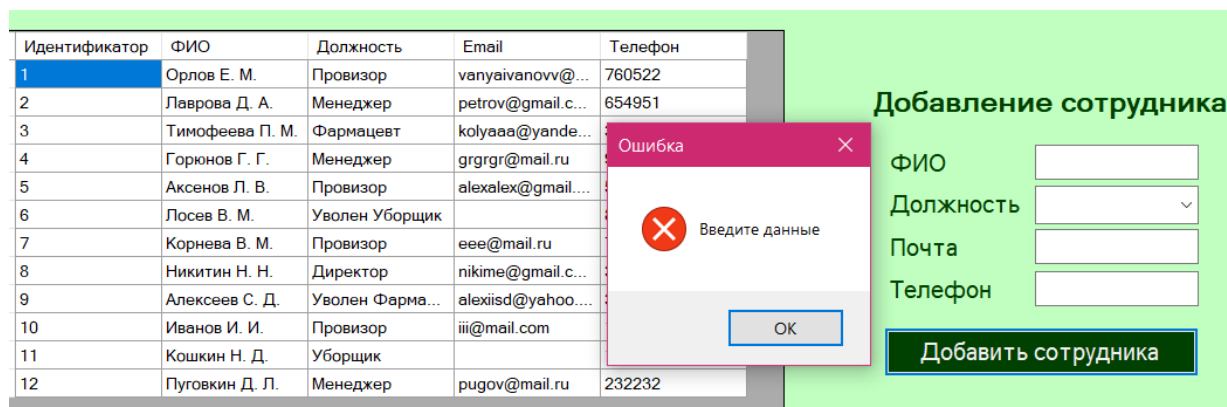


Рис.2.2.42 Проверка введенных данных

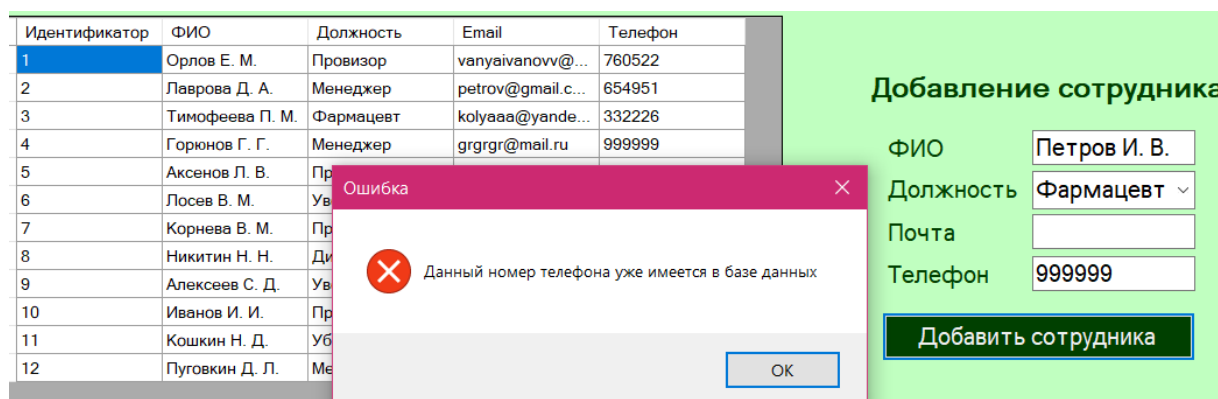


Рис.2.2.43 Проверка уникальности телефона

Также на форме учета сотрудников есть функция увольнения, которая реализована с помощью хранимой процедуры retireSotr (Приложение А, п.17). Для увольнения сотрудника требуется выбрать в таблице строку с нужным сотрудником и нажать кнопку «Уволить сотрудника» (рис.2.2.44). После увольнения к должности припишется в начале «Уволен».

Идентификатор	ФИО	Должность	Email	Телефон
1	Орлов Е. М.	Провизор	vanyaivanovv@...	760522
2	Лаврова Д. А.	Менеджер	petrov@gmail.c...	654951
3	Тимофеева П. М.	Фармацевт	kolyaaa@yande...	332226
4	Горюнов Г. Г.	Менеджер	grgrgr@mail.ru	999999
5	Аксенов Л. В.	Провизор	alexalex@gmail...	574666
6	Лосев В. М.	Уволен Уборщик		852304
7	Корнева В. М.	Провизор	eee@mail.ru	741560
8	Никитин Н. Н.	Директор	nikime@gmail.c...	333333
9	Алексеев С. Д.	Уволен Фармацевт	alexiiisd@yahoo...	369149
10	Иванов И. И.	Провизор	iii@mail.com	123456
11	Кошкин Н. Д.	Уборщик		123654
12	Пуговкин Д. Л.	Уволен Менеджер	pugov@mail.ru	232232

Добавление сотрудника

ФИО

Должность

Почта

Телефон

Выберите сотрудника в таблице

Рис.2.2.44 Увольнение сотрудника

Вернемся к форме авторизации. Так как у аптеки есть директор, нужно, чтобы у него был доступ ко всем функциям, которые имеются и у первостольника, и у менеджера. Поэтому, при авторизации директора по своим логину и паролю, он попадает на форму «Директор», где есть три кнопки: «Учет товаров» и «Учет сотрудников» перенаправляют на соответствующие формы, которые доступны менеджеру, а кнопка «Касса» направляет на форму первостольника.

Директор

Выберите, с чем хотите работать

Рис.2.2.45 Форма директора

ЗАКЛЮЧЕНИЕ

Тема курсового проекта – проектирование базы данных для упрощения работы аптеки и для автоматизации процессов совершения закупок, продаж, списания товаров. Этот проект актуален, так как позволяет обрабатывать большое количество информации, осуществлять быстрый поиск информации, управлять складом, продажами, следить за состоянием товаров.

В ходе выполнения работы были получены следующие результаты:

1. Изучена предметная область, проанализированы примеры реализации подобных программных продуктов – программа 1С:Аптека от компании 1С и конфигурация «Аптека» от компании «Простой софт». Выделены основные функции, аспекты будущей системы.

2. На этапе концептуального проектирования выделены основные действующие лица БД: первостольник и менеджер. Определены их функции, и в результате этого составлена диаграмма вариантов использования. По результатам концептуального проектирования выделены основные сущности, информация о которых будет храниться в БД.

3. По результатам предыдущего этапа построены ER-диаграммы для объектов, которые будут храниться в БД, определены типы связей между сущностями и классы принадлежности. По данным типам связи и классам принадлежности сформированы отношения. Во все полученные отношения (Товар, Препарат, Категория, Сотрудник, Продажа, Поставщик, Закупка, СоставЗакупки, СписанныйТовар, СоставПродажи, Аналоги) добавлены не ключевые атрибуты.

4. На этапе физического проектирования в среде SQL Server Management Studio были созданы таблицы и связи между ними. Для обеспечения полноценного функционирования программной части были разработаны хранимые процедуры и функции.

5. На заключительном этапе был разработан пользовательский интерфейс для взаимодействия с базой данных с помощью Windows Forms в среде разработки Visual Studio на языке программирования C#. На основе

данных, которые были определены в процессе проектирования, были разработаны БД и интерфейс взаимодействия с ней, позволяющие выполнить различные функции в зависимости от роли пользователя. Для Первостольника доступны следующие функции: поиск по товарам, получение информации о ценах и количестве товара в наличии, оформление продаж, подбор аналогов, просмотр характеристик препаратов. Менеджеру доступны следующие функции: добавление и увольнение сотрудников, совершение закупок, получение информации о наличии товара в аптеке, списание товаров, поиск по товарам и получение информации о них, контроль продаж, учёт поставщиков.

В итоге был получен программный продукт для оптимизации работы аптеки, полностью соответствующий задачам, поставленным на этапе проектирования.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Сайт «Интуит» [Электронный ресурс] – Режим доступа: <https://intuit.ru/studies/courses/574/430/lecture/9749#>, свободный. Дата обращения: 24.03.2023 г.
2. Сайт «Live Medical» [Электронный ресурс] – Режим доступа: <https://www.livemedical.ru/tools/apteka/>, свободный. Дата обращения: 24.03.2023 г.
3. Сайт «1С Платформа IT» [Электронный ресурс] – Режим доступа: <https://1c.platforma-it.ru/catalog/otrasl/torgovlya/1c-apteka/>, свободный. Дата обращения: 25.03.2023 г.
4. Сайт «Викисловарь» [Электронный ресурс] – Режим доступа: <https://ru.wiktionary.org/wiki/>, свободный. Дата обращения: 25.03.2023 г.
5. Сайт «Простой софт» [Электронный ресурс] – Режим доступа: <https://prostoysoft.ru/Pharmacy.htm>, свободный. Дата обращения: 25.03.2023 г.
6. Журнал «Успехи современного естествознания» [Электронный ресурс] – Режим доступа: <https://natural-sciences.ru/ru/article/view>, свободный. Дата обращения: 11.04.2023 г.
7. Сайт «Витебская академия ветеринарной медицины» [Электронный ресурс] – Режим доступа: <https://www.vsavm.by/knigi/kniga3/1020.html>, свободный. Дата обращения: 15.05.2023 г.
8. Сайт «Falcon space» [Электронный ресурс] – Режим доступа: <https://falconspace.ru/list/sqlserver/ustanovka-sql-server-management-studio---488#>, свободный. Дата обращения: 16.05.2023 г.
9. Сайт «Learn Microsoft», Техническая документация по SQL Server [Электронный ресурс] – Режим доступа: <https://learn.microsoft.com/sql/sql-server/?view=sql-server-ver16>, свободный. Дата обращения: 16.05.2023 г.
10. Сайт «Learn Microsoft», Документация по Windows Forms [Электронный ресурс] – Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/desktop/winforms/?view=netdesktop-8.0>, свободный. Дата обращения: 16.05.2023 г.

11. Сайт «Learn Microsoft», Документация по интегрированной среде разработки Visual Studio [Электронный ресурс] – Режим доступа: <https://learn.microsoft.com/ru-ru/visualstudio/ide/?view=vs-2022>, свободный. Дата обращения: 25.11.2023 г.

12. Сайт «Learn Microsoft», Документация по C# [Электронный ресурс] – Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/csharp/>, свободный. Дата обращения: 16.05.2023 г.

SQL-операторы создания программных объектов БД

1. Создание хранимой процедуры «authenticationUser»

```
CREATE PROCEDURE [dbo].[authenticationUser]
    @fio VARCHAR(60),
    @phoneNum VARCHAR(15),
    @a INT OUTPUT
AS
BEGIN
    SET @a = 0

    declare @position VARCHAR(40)
    SET @position = (SELECT Position FROM Sotr WHERE @fio = Fio AND
@phoneNum = PhoneNum)

    IF @position = 'Провизор' OR @position = 'Фармацевт'
    SET @a = 1

    IF @position = 'Менеджер'
    SET @a = 2

    IF @position = 'Директор'
    SET @a = 3
END
```

2. Создание представления «tableProduct»

```
CREATE VIEW [dbo].[tableProduct]
AS
SELECT dbo.Product.Barcode, dbo.Category.NameCategory, dbo.Product.Brand,
dbo.Product.NameProduct, dbo.Product.Price, dbo.Product.MinReserve,
dbo.Product.InStock
FROM    dbo.Category INNER JOIN
        dbo.Product ON dbo.Category.CodeCategory =
dbo.Product.CodeCategory
```

3. Создание хранимой процедуры «addStructSell»

```
CREATE PROCEDURE [dbo].[addStructSell]
    @barcode CHAR(13),
    @quantity SMALLINT,
    @sotr VARCHAR(60)
AS
```

```

BEGIN
    BEGIN
        IF (SELECT InStock FROM Product
            WHERE Barcode = @barcode) > 0
        BEGIN
            IF (SELECT InStock FROM Product
                WHERE Barcode = @barcode) >= @quantity
            BEGIN
                DECLARE @idSotr smallint
                SELECT @idSotr = idSotr FROM Sotr
                WHERE @sotr = Fio

                DECLARE @price MONEY
                SELECT @price = Price FROM Product

                DECLARE @total MONEY
                SET @total = @price * @quantity

                DECLARE @n_receipt SMALLINT
                SELECT @n_receipt = (MAX(Nrec) + 1)
                FROM StructSell

                IF NOT EXISTS (SELECT * FROM Sell
                    WHERE Nreceipt = @n_receipt)
                BEGIN
                    INSERT INTO Sell (Nreceipt, id_sotr, DateSell)
                    VALUES (@n_receipt, @idSotr, GETDATE())
                END

                INSERT INTO StructSell(Nrec, Barcode, Quantity,
TotalCost)
                VALUES (@n_receipt, @barcode, @quantity, @total)
            END
        ELSE BEGIN
            PRINT 'Товара недостаточно'
        END
        UPDATE Product SET InStock = (InStock - @quantity)
        WHERE Barcode = @barcode
    END
ELSE BEGIN
    PRINT 'Товара нет в наличии'
END
END
END

```

4. Создание табличной функции «seeDescriptionOfMedicament»

```
CREATE FUNCTION [dbo].[seeDescriptionOfMedicament]
(@barcode CHAR(13))
RETURNS TABLE
AS
RETURN
(
    SELECT PharmGroup AS 'Фарм.группа',
           Exp AS 'Срок годности',
           Notes AS 'Примечания',
           Effect AS 'Эффект',
           ActiveSubstance AS 'Действующее вещество',
           ReleaseForm AS 'Форма выпуска',
           Dosage AS 'Дозировка',
           Quantity AS 'Количество',
           Volume AS 'Объем',
           Composition AS 'Состав',
           Country AS 'Страна',
           Manufactur AS 'Производитель'
    FROM Medicament WHERE Barcode = @barcode
)
```

5. Создание скалярной функции «getProductInfo»

```
CREATE FUNCTION [dbo].[getProductInfo] (@barcode CHAR(13))
RETURNS VARCHAR(205)
AS
BEGIN
    DECLARE @productInfo VARCHAR(205)

    SELECT @productInfo = CONCAT(Brand, ' ', NameProduct, ' ', Price, ' руб')
    FROM Product WHERE Barcode = @barcode

    RETURN @productInfo
END
```

6. Создание табличной функции «findAnalog»

```
CREATE FUNCTION [dbo].[findAnalog] (@Barcode CHAR(13))
RETURNS TABLE
AS
RETURN
(
    SELECT Product.Barcode AS 'Штрихкод', Brand AS 'Бренд', NameProduct AS
    'Название', Price AS 'Цена', InStock AS 'В наличии'
    FROM Product
)
```

```

INNER JOIN Analog ON Product.Barcode = Analog.BarcodeAnalog
WHERE Analog.Barcode = @Barcode
)

```

7. Создание табличной функции «getPurveyor»

```

CREATE FUNCTION [dbo].[getPurveyor]()
RETURNS TABLE
AS
RETURN ( SELECT NamePurveyor FROM Purveyor ))

```

8. Создание хранимой процедуры «addPurchase»

```

CREATE PROCEDURE [dbo].[addPurchase]
    @barcode CHAR(13),
    @quantity SMALLINT,
    @purveyor varchar(20),
    @sotr VARCHAR(60)
AS
BEGIN
    --по имени поставщика вытягивается его код и вносится в таблицу
    поставок
    DECLARE @codePurveyor smallint
    SELECT @codePurveyor = CodePurveyor FROM Purveyor WHERE
    @purveyor = NamePurveyor

    -- есть ли в таблице Purveyor такой поставщик
    IF NOT EXISTS(SELECT NamePurveyor FROM Purveyor WHERE
    @purveyor = NamePurveyor)
    BEGIN
        RAISERROR ('NamePurveyor does not exist in Purveyor table', 16,
        1);
        RETURN;
    END

    --по имени сотрудника вытягивается его код и вносится в таблицу
    поставок
    DECLARE @idSotr smallint
    SELECT @idSotr = idSotr FROM Sotr WHERE @sotr = Fio

    DECLARE @price MONEY
    SELECT @price = Price FROM Product WHERE Barcode = @barcode

    DECLARE @total MONEY
    SET @total = @price * @quantity

```

```

DECLARE @n_ord SMALLINT
SELECT @n_ord = (MAX(OrderCode) + 1) FROM StructPurchase

IF NOT EXISTS (SELECT * FROM Purchase
WHERE OrderCode = @n_ord)
BEGIN
    INSERT INTO Purchase(OrderCode, CodePurv, id_sotr, DateExe,
DelivPrice)
    VALUES (@n_ord, @codePurveyor, @idSotr, GETDATE(), @total)
END

INSERT INTO StructPurchase(OrderCode, Barcode, ValidFor, UnitPrice,
QuantityOrd)
    VALUES (@n_ord, @barcode, NULL, @price, @quantity)

UPDATE Product SET InStock = (InStock + @quantity)
WHERE Barcode = @barcode
END

```

9. Создание хранимой процедуры «addPurveyor»

```

CREATE PROCEDURE [dbo].[addPurveyor]
    @namePurveyor varchar(20),
    @contactPerson varchar(50),
    @country varchar(30),
    @sity varchar(30),
    @address varchar(100),
    @phoneNum varchar(20),
    @termsPay varchar(100)
AS
BEGIN
    DECLARE @codePurv smallint
    SELECT @codePurv = (MAX(CodePurveyor) + 1) FROM Purveyor

    INSERT INTO Purveyor(CodePurveyor, NamePurveyor, ContactPerson,
Country, Sity, Address, PhoneNum, TermsPay)
    VALUES (@codePurv, @namePurveyor, @contactPerson, @country,
@sity, @address, @phoneNum, @termsPay)
END

```

10. Создание хранимой процедуры «addNewProduct»

```

CREATE PROCEDURE [dbo].[addNewProduct]
(
    @barcode CHAR(13),
    @nameCategory VARCHAR(30),

```

```

@brand VARCHAR(30),
@nameProduct VARCHAR(170),
@price MONEY,
@minReserve SMALLINT,
@inStock SMALLINT
)
AS
BEGIN
    DECLARE @codeCategory TINYINT
    SELECT @codeCategory = CodeCategory FROM Category WHERE
@nameCategory = NameCategory

    INSERT INTO Product (Barcode, CodeCategory, Brand, NameProduct, Price,
MinReserve, InStock)
    VALUES (@barcode, @codeCategory, @brand, @nameProduct, @price,
@minReserve, @inStock)
END

```

11. Создание хранимой процедуры «addMedicament»

```

CREATE PROCEDURE [dbo].[addMedicament]
(
    @barcode CHAR(13),
    @pharmGroup varchar(50),
    @exp VARCHAR(20),
    @notes VARCHAR(150),
    @effect VARCHAR(50),
    @activeSubstance VARCHAR(70),
    @releaseForm VARCHAR(80),
    @dosage VARCHAR(50),
    @quantity TINYINT,
    @volume VARCHAR(15),
    @composition VARCHAR(400),
    @country VARCHAR(30),
    @manufactur VARCHAR(70)
)
AS
BEGIN
    DECLARE @codeMedicament SMALLINT
    SELECT @codeMedicament = (MAX(CodeMedicament) + 1) FROM
Medicament

    INSERT INTO Medicament (CodeMedicament, Barcode, PharmGroup,
Exp, Notes, Effect, ActiveSubstance, ReleaseForm, Dosage, Quantity, Volume,
Composition, Country, Manufactur)

```

```
VALUES (@codeMedicament, @barcode, @pharmGroup, @exp, @notes,
@effect, @activeSubstance, @releaseForm, @dosage, @quantity, @volume,
@composition, @country, @manufactur)
END
```

12. Создание хранимой процедуры «writeOffProduct»

```
CREATE PROCEDURE [dbo].[writeOffProduct]
    @Barcode char(13),
    @Quantity smallint,
    @WhyDelete varchar(60)
AS
BEGIN
    DECLARE @Ndelete smallint
    DECLARE @DateDelete date
    DECLARE @InStock smallint

    SELECT @Ndelete = (MAX(Ndelete) + 1) FROM DeleteProduct

    SELECT @InStock = InStock FROM Product
        WHERE Barcode = @Barcode;

    SET @DateDelete = GETDATE()

    IF @Quantity <= @InStock
    BEGIN
        INSERT INTO DeleteProduct (Ndelete, Barcode, Quantity,
DateDelete, WhyDelete)
            VALUES (@Ndelete, @Barcode, @Quantity, @DateDelete,
@WhyDelete)

        UPDATE Product
        SET InStock = InStock - @Quantity
        WHERE Barcode = @Barcode
    END
    ELSE
    BEGIN
        PRINT 'Такое количество товара отсутствует на складе'
    END
END
```

13. Создание представления «deleteProductForWF»

```
CREATE VIEW [dbo].[deleteProductForWF] AS
```

```

SELECT dbo.DeleteProduct.Ndelete, dbo.DeleteProduct.Barcode,
dbo.Product.Brand, dbo.Product.NameProduct, dbo.DeleteProduct.Quantity,
dbo.DeleteProduct.DateDelete, dbo.DeleteProduct.WhyDelete
FROM    dbo.DeleteProduct INNER JOIN
        dbo.Product ON dbo.DeleteProduct.Barcode = dbo.Product.Barcode

```

14. Создание представления «viewSell»

```

CREATE VIEW [dbo].[viewSell] AS
SELECT dbo.Sell.Nreceipt, dbo.Sotr.Fio, dbo.Sell.DateSell,
dbo.StructSell.Barcode, dbo.Product.Brand, dbo.Product.NameProduct,
dbo.StructSell.Quantity, dbo.StructSell.TotalCost
FROM    dbo.Sotr INNER JOIN
        dbo.Sell ON dbo.Sotr.idSotr = dbo.Sell.id_sotr INNER JOIN
        dbo.StructSell ON dbo.Sell.Nreceipt = dbo.StructSell.Nrec INNER
JOIN
        dbo.Product ON dbo.StructSell.Barcode = dbo.Product.Barcode

```

15. Создание представления «viewPurchase»

```

CREATE VIEW [dbo].[viewPurchase] AS
SELECT dbo.Purchase.OrderCode, dbo.Purveyor.NamePurveyor, dbo.Sotr.Fio,
dbo.Purchase.DateExe, dbo.Purchase.DelivPrice, dbo.StructPurchase.Barcode,
dbo.Product.Brand, dbo.Product.NameProduct, dbo.StructPurchase.UnitPrice,
        dbo.StructPurchase.QuantityOrd
FROM    dbo.Purveyor INNER JOIN
        dbo.Purchase ON dbo.Purveyor.CodePurveyor =
dbo.Purchase.CodePurv INNER JOIN
        dbo.Sotr ON dbo.Purchase.id_sotr = dbo.Sotr.idSotr INNER JOIN
        dbo.StructPurchase ON dbo.Purchase.OrderCode =
dbo.StructPurchase.OrderCode INNER JOIN
        dbo.Product ON dbo.StructPurchase.Barcode = dbo.Product.Barcode

```

16. Создание хранимой процедуры «addNewSotr»

```

CREATE PROCEDURE [dbo].[addNewSotr]
(
    @fio VARCHAR(60),
    @position VARCHAR(40),
    @email VARCHAR(50),
    @phoneNum VARCHAR(15)
)
AS
BEGIN
    DECLARE @maxIdSotr SMALLINT;
    -- Получаем максимальное значение idSotr из таблицы Sotr
    SELECT @maxIdSotr = MAX(idSotr) FROM Sotr;

```



```
INSERT INTO Sotr (idSotr, Fio, Position, Email, PhoneNum)
VALUES (@maxIdSotr + 1, @fio, @position, @email, @phoneNum)
END
```

17. Создание хранимой процедуры «retireSotr»

```
CREATE PROCEDURE [dbo].[retireSotr]
    @idSotr SMALLINT
AS
BEGIN
    UPDATE Sotr
    SET Position = 'Уволен ' + Position
    WHERE idSotr = @idSotr;
END
```

18. Создание скалярной функции «checkStock»

```
CREATE FUNCTION [dbo].[checkStock] (@barcode CHAR(13))
RETURNS INT
AS
BEGIN
    DECLARE @res int
    SELECT @res = InStock FROM Product WHERE Barcode = @barcode
    RETURN @res
END
```

19. Создание скалярной функции «checkBarcode»

```
CREATE FUNCTION [dbo].[checkBarcode] (@barcode CHAR(13))
RETURNS INT
AS
BEGIN
    DECLARE @res int
    SELECT @res = COUNT(*) FROM Product
    WHERE Barcode = @barcode
    RETURN @res
END
```

ПРИЛОЖЕНИЕ Б

Заполнение таблиц данными

Barcode	CodeCategory	Brand	NameProduct	Price	MinReserve	InStock
0023456789004	9	Armed	концентратор кислорода 8f-1/белый	26289,00...	1	21
0023456789031	9	AMRUS	Термометр медицинский инфракрасный amrus amit-120	1250,0000	20	30
0023456789071	10	Б.Браун Медикал	Эпиген интим 0,1% спрей 15 мл	1902,0000	5	35
0023456789076	11	SFM HOSPITAL	Шприц 50 мл 3-х комп с иглой 18g 25 шт	862,0000	50	84
0123456789009	11	DERMAGRIP	Перчатки смотровые dermagrip high risk неопудренные s 25 пар	1095,0000	30	88
0123456789036	10	Amra	диски ватные 180 шт.	116,0000	30	84
0723456789023	5	Реневал	Парацетамол	50,0000	40	262
0812345678958	1	КОМФОРТ	Корректор осанки КОМФ-ОПТ /арт. К-506/размер М/рост 160-170	1250,0000	5	30
1000987654325	8	Maxima	55 comfort + контактные линзы плановой замены 6 шт.	941,0000	20	3
1001234567898	8	OPTICLEAN	Opticlean салфетки влажные для оптики 20 шт.	42,0000	10	14
1009876543213	8	Гельтек-Медика	Блефаросалфетка салфетка влажная стерильная для ухода за век...	505,0000	5	29
1123456789014	9	OMRON	Термометр omron eco temp basic mc-246-ru электронный	563,0000	30	109
1123456789019	10	Kleenex	антибактериальные влажные салфетки 40 шт.	174,0000	20	58
1123456789050	11	INEKTA	Катетер фoley латексный с силиконовым покрытием 2-ходовый ...	500,0000	10	20
1123456789055	10	Тена	lady slim ultra mini урологические прокладки 14 шт.	217,0000	10	33
1123456789091	9	AND	Ингалятор CN-233 компрессорный	2535,0000	5	21
1223456789096	11	SFM	ПЕРЧАТКИ СМОТР SFM НИТРИЛ Н/СТЕР НЕОПУДР ТЕКСТУР ВНУТ...	1022,0000	30	60
1234567890125	2	Фармстандарт	Фарингазон 10 мг 20 шт. таблетки для рассасывания	190,0000	10	30
1234567891124	2	Коделак	Сироп от кашля коделак бронхо с чабрецом 400 мл	600,0000	5	55
1234567891234	2	Тева	Спрей назальный ринонорм 0,05	150,0000	10	0
1236547895123	1	Уреа	капли назальные 0,1 мг	450,0000	4	0
1596324788741	10	Lactacyd	салфетки для интимной гигиены 15 шт.	106,0000	0	142
1823456789007	11	МПК Елец	Шприц 50 мл трехдетальный стер однократ прим луер с иглой 1,...	1011,0000	50	147
1912345678931	1	ИНТЕКС	Бандаж на коленный сустав наколенник интекс класс 1/размер I/...	605,0000	5	40
2000987654324	8	Alcon	precision1 однодневные контактные линзы 8,3/14,2 30 шт.	1637,0000	20	160
2001234567897	8	Avizor	aqua soft раствор для линз 120 мл	400,0000	5	25
2009876543216	8	АМТ ТРЕЙД	Пипетка офтальмологическая в футляре/в индивидуальной упак...	18,0000	5	20
2012345678908	3	IME-DC	Ланцеты ime-dc 100 шт	350,0000	10	80
2023202320232	1	Артека	стельки	2500,0000	1	90
2223456789001	10	Mon rulon	бумага туалетная влажная 80 шт.	98,0000	10	30
2223456789006	9	AND	Термометр DT-623 электронный с гибким наконечником	408,0000	50	98
2223456789033	11	ТЕРМО-КОНТ М...	Термоконтанер термо-конт мк тм-1 многоразового использова...	2155,0000	5	60

Рис.Б.1. Данные таблицы «Товар»

C...	Barcode	PharmGroup	Exp	Notes	Effect	ActiveSubs...	ReleaseForm	Dosage	Quantity	Volume	Composition	Country	Manufactur
1	002345...	противови...	Более 12 месяцев	Беречь от детей	противовирусно...	Аммония г...	спрей	0,1%	NULL	15 мл	Глицирризиновая кислота акт...	Испан...	Б.Браун Медикал С.А.
2	072345...	анальгезир...	Более 12 месяцев	Беречь от детей	обезболивающее	Парацетам...	таблетки	0,5	10	NULL	парацетамол - 500,0 мг; вспо...	Россия	Обновление ПОК АО
3	123456...	антисептич...	Более 12 месяцев	Детям с 6-ти лет	антисептик	Амбазон	таблетки дл...	0,01	20	NULL	амбазона моногидрат - 10,0 ...	Россия	Фармстандарт-Лекс...
4	123456...	отхаркива...	Более 12 месяцев	Беречь от детей	отхаркивающее	Натрия гли...	эликсир	NULL	NULL	100Млч	амброксола гидрохлорид - 10...	Россия	Фармстандарт-Лекс...
5	123456...	противоко...	Более 12 месяцев	Беречь от детей	сосудосуживаю...	Ксиломета...	спрей назал...	0,1Мг+...	NULL	10МЛ	ксилометазолина гидрохлори...	Герма...	Меркле ГмБХ
6	978020...	Аллергия	Более 12 месяцев	Беречь от детей	Антигистаминное	Диметинден	капли для п...	1 Мг/М...	NULL	30 мл	диметиндена малеат 1,0 мг, Н...	Швейц...	ГСК Консьюмер Хел...
7	332345...	противови...	Более 12 месяцев	Беречь от детей	противовирусное	Аммония г...	спрей	0,1%	NULL	15 мл	Аммония глицирризинат - 0...	Россия	Вифитех ЗАО
8	412345...	анальгезир...	От 6 до 12 меся...	Беречь от детей	обезболивающее	Фенпивери...	таблетки	NULL	10	NULL	метамизол натрия 500 мг, пит...	Болга...	Балканфарма - Дупн...
9	641234...	гипоглике...	Более 12 месяцев	Хранить в хол...	гипогликемичес...	Инсулин гл...	раствор для...	100ЕД/...	5	3 мл	инсулин гларгин - 100 ЕД (3,6...	Россия	ГЕРОФАРМ ООО
10	751234...	антисептич...	Более 12 месяцев	Беречь от детей	антисептик для г...	Амбазон	таблетки дл...	10 мг	10	NULL	Амбазона моногидрат 10 мг В...	Румын...	С.К.Терапия С.А.
11	852345...	нестероид...	Более 12 месяцев	Хранить в защ...	обезболивающее	Нимесулид	таблетки	100 мг	20	NULL	нимесулид 100 мг.Вспомогате...	Индия	Д-р Редди'с Лаборат...
12	861234...	антисептич...	Более 12 месяцев	Беречь от детей	антисептик для г...	Лизоцим Ц...	спрей для м...	0,0001...	NULL	NULL	Лидокаина гидрохлорида мо...	Босни...	Босналек АО
13	871234...	кератопрот...	Более 12 месяцев	Хранить в хол...	NULL	Пластохин...	капли глазн...	0,155М...	NULL	5МЛ	Пластохинонилдецилтрифени...	Россия	Фирн М ООО/НИИ ...
14	951632...	поливитам...	Более 12 месяцев	Хранить в сухо...	NULL	Витамин ЕР...	капсулы	100 мг ...	10	NULL	альфа-токоферола ацетата (в...	Россия	Мелиген ФП ЗАО
15	971234...	Медиздели...	Более 12 месяцев	Беречь от детей	NULL	Гиалуронат...	шприц	1,5%	1	5,0 мл	Гиалуроновая кислота (гиалу...	Италия	Фидиа Фармацеути...
16	978020...	противоал...	Более 12 месяцев	Беречь от детей	антигистаминное	Диметинден	капли для п...	1 Мг/М...	NULL	20 мл	диметиндена малеат 1,0 мг.1 ...	Швейц...	ГСК Консьюмер Хел...

Рис.Б.2. Данные таблицы «Препарат»

CodeCategory	NameCategory
1	Ортопедические товары
2	Ухо, горло, нос
3	Диабет
4	Болезни суставов
5	Обезболивающие
6	Аллергия
7	Витамины и добавки
8	Зрение
9	Медицинские приборы
10	Гигиена
11	Медицинские изделия

Рис.Б.3. Данные таблицы «Категория»

idSotr	Fio	Position	Email	PhoneNum
1	Орлов Е. М.	Провизор	vanyaivanovv@mail.ru	760522
2	Лаврова Д. А.	Менеджер	petrov@gmail.com	654951
3	Тимофеева П. М.	Фармацевт	kolyaaa@yandex.ru	332226
4	Горюнов Г. Г.	Менеджер	grgrgr@mail.ru	999999
5	Аксенов Л. В.	Провизор	alexalex@gmail.com	574666
6	Лосев В. М.	Уволен Уборщик		852304
7	Корнева В. М.	Провизор	eee@mail.ru	741560
8	Никитин Н. Н.	Директор	nikime@gmail.com	333333
9	Алексеев С. Д.	Уволен Фармацевт	alexiisd@yahoo.com	369149
10	Иванов И. И.	Провизор	iii@mail.com	123456
11	Кошкин Н. Д.	Уборщик		123654

Рис.Б.4. Данные таблицы «Сотрудник»

Nreceipt	id_sotr	DateSell
1	1	2022-10-02
2	5	2022-11-03
3	6	2022-12-12
4	7	2023-01-29
5	7	2023-02-15
6	5	2023-03-30
7	5	2023-04-08
8	1	2023-05-16
9	5	2023-06-01
10	10	2023-12-12
11	10	2023-12-12
12	10	2023-12-12
13	10	2023-12-12
14	10	2023-12-12
15	10	2023-12-12
16	10	2023-12-12
17	10	2023-12-12
18	10	2023-12-20
19	10	2023-12-20
20	10	2023-12-20
21	10	2023-12-20
22	10	2023-12-20
23	10	2023-12-20
24	10	2023-12-20
25	1	2023-12-20
26	7	2023-12-20
27	7	2023-12-20
28	7	2023-12-20
29	7	2023-12-20
30	7	2023-12-20
31	7	2023-12-20

Рис.Б.5. Данные таблицы «Продажа»

CodePurveyor	NamePurveyor	ContactPerson	Country	Sity	Address	PhoneNum	TermsPay
1	Аделейд	Казаков Б.Л.	Беларусь	Минск	ул.Стрелецкая, 9	788369	безнал
2	ПрофитМед	Сидоренко А.В.	Россия	Москва	ул.Мира, 1	742330	NULL
3	НПО Гарант	Кошка А.Е.	Россия	Рязань	ул.Дружбы, 3	541205	безнал
4	МедТорг	Румянцев И.И.	Россия	Белгород	ул.Энгельса, 16	741568	наличные
5	ООО МедФарм	Зорьин А.Г.	Россия	Химки	ул.Часовая, 9	850023	безнал
6	ООО Север	Овчаров Д.А.	Украина	Киев	ул.Ленина, 46	852001	корр.счет 46...
7	ИП Медичек	Рыбкин П.Б.	Россия	Москва	ул.Юных Ленинцев, 103	856910	NULL
8	PHARMA Group	Сорокина С. Д.	Россия	Подольск	ш.Ломоносова, 24	196355	NULL
9	МФК «Биоритм»	Филиппова Е.Т.	Россия	Ступино	ул.Бухарестская, 54	747665	безнал
10	Orion Pharma	Nieminen Ilma	Финляндия	Эспоо	Siltakatu 13	852223	корр.счёт 26...

Рис.Б.6. Данные таблицы «Поставщик»

OrderCode	CodePurv	id_sotr	DateExe	DelivPrice
10	10	4	2023-12-20	6150,0000
11	10	4	2023-12-20	20500,0000
12	10	4	2023-12-20	6150,0000
13	1	4	2023-12-20	16400,0000
14	1	4	2023-12-20	16400,0000
15	1	4	2023-12-20	8200,0000
16	1	4	2023-12-20	16400,0000
17	1	4	2023-12-20	16400,0000
18	1	4	2023-12-20	12300,0000
19	3	4	2023-12-20	12300,0000
20	3	8	2023-12-20	8200,0000
21	3	8	2023-12-20	8200,0000
22	3	8	2023-12-20	20500,0000
23	3	8	2023-12-20	2050,0000
24	3	8	2023-12-20	10250,0000
25	3	8	2023-12-20	8200,0000
26	3	8	2023-12-20	2050,0000
27	1	4	2023-12-20	2050,0000
28	1	4	2023-12-20	410,0000
29	7	4	2023-12-20	8200,0000
30	7	4	2023-12-20	5630,0000
31	8	4	2023-12-20	3080,0000
32	4	4	2023-12-20	6150,0000
33	1	4	2023-12-20	10100,0000
34	6	4	2023-12-20	8000,0000
35	6	4	2023-12-20	13680,0000
36	2	4	2023-12-20	1770,0000
37	2	4	2023-12-20	42345,0000
38	7	4	2023-12-20	28450,0000
39	7	4	2023-12-20	2845,0000
40	7	4	2023-12-20	6250,0000

Рис.Б.7. Данные таблицы «Закупка»

Ndelete	Barcode	Quantity	DateDelete	WhyDelete
1	6323456789093	2	2022-09-01	Истекает срок годности
2	0023456789031	5	2023-12-10	Истекает срок годности
3	0023456789071	10	2023-12-10	Истекает срок годности
4	0023456789071	2	2023-12-10	Истекает срок годности
5	0023456789071	3	2023-12-10	Истекает срок годности
6	0123456789036	16	2023-12-19	Товар испорчен
7	1000987654325	6	2023-12-19	Истекает срок годности
8	9923456789058	6	2023-12-19	Истекает срок годности
9	1123456789014	6	2023-12-19	Истекает срок годности
10	0023456789076	16	2023-12-19	Товар испорчен
11	0723456789023	8	2023-12-19	Истекает срок годности
12	0723456789023	3	2023-12-19	Истекает срок годности
13	1001234567898	3	2023-12-19	Нарушены условия хранения
14	0812345678958	5	2023-12-20	Истекает срок годности
15	1000987654325	15	2023-12-20	Истекает срок годности
16	1000987654325	1	2023-12-20	Истекает срок годности

Рис.Б.8. Данные таблицы «СписанныйТовар»

OrderCode	Barcode	ValidFor	UnitPrice	QuantityOrd
28	1009876543213	2029-01-01	410,0000	1
29	5223456789006	2029-01-01	410,0000	20
30	1123456789014	2026-01-01	563,0000	10
31	7723456789010	2026-01-01	154,0000	20
32	9923456789099	2025-01-01	410,0000	15
33	1009876543213	2025-01-01	505,0000	20
34	9923456781462	2029-01-01	400,0000	20
35	9923456789017	2026-01-01	456,0000	30
36	5523456789095	2030-01-01	59,0000	30
37	1000987654325	2040-01-01	941,0000	45
38	8712345678902	2026-01-01	569,0000	50
39	8712345678902	2025-01-01	569,0000	5
40	0023456789031	2026-01-01	1250,0000	5
41	0023456789004	2029-01-01	26289,0000	5
42	0723456789023	2026-01-01	50,0000	50
43	1823456789007	2029-01-01	1011,0000	50
44	7823456789097	2029-01-01	197,0000	40
45	7823456789097	2030-01-01	197,0000	30
46	3323456789093	2040-01-01	1742,0000	50
47	1000987654325	2029-01-01	941,0000	30
48	5523456789059	2026-01-01	1051,0000	15
49	4123456789020	2029-01-01	195,0000	100
50	0812345678958	2040-01-01	1250,0000	20
51	5523456789090	2025-01-01	789,0000	25
52	7723456789051	2030-01-01	388,0000	50
53	1912345678931	2040-01-01	605,0000	30
54	4423456789030	2029-01-01	475,0000	45
55	8711234567893	2026-01-01	1168,0000	10

Рис.Б.9. Данные таблицы «СоставЗакупки»

Nrec	Barcode	Quantity	TotalCost
2	1123456789019	2	348,0000
2	6323456789093	2	550,0000
3	0023456789004	2	52578,0000
4	1009876543213	1	505,0000
5	1123456789014	1	563,0000
6	0723456789023	2	100,0000
7	0023456789076	1	862,0000
8	0023456789004	1	26289,0000
9	2001234567897	1	400,0000
10	1123456789014	2	1126,0000
11	6323456789093	2	550,0000
12	1009876543213	2	1010,0000
13	0123456789009	2	2190,0000
14	9923456789012	1	7267,0000
15	8612345678909	3	1200,0000
16	9923456789012	2	14534,0000
17	9923456789012	2	14534,0000
18	1823456789007	3	3033,0000
19	8523456789058	3	810,0000
20	4523456789035	12	852,0000
21	9023456789053	2	562,0000
22	9516328741592	4	172,0000
23	1123456789055	7	1519,0000
24	8923456789079	1	137,0000
25	4212345678976	4	6000,0000
26	4123456789020	7	1365,0000
27	2223456789006	2	816,0000
28	0723456789023	2	100,0000
29	0723456789023	5	250,0000
30	5312345678950	1	1850,0000

Рис.Б.10. Данные таблицы «СоставПродажи»

Barcode	BarcodeAnalog
0123456789009	1223456789096
0123456789036	3323456789098
1000987654325	8712345678904
1001234567898	5909876543210
1123456789014	2223456789006
1123456789019	8923456789079
1123456789050	2923456789094
1123456789091	2223456789073
1223456789096	0123456789009
1823456789007	0023456789076
2000987654324	5901234567895
2001234567897	8712345678906
2223456789001	5523456789054
2223456789006	1123456789014
2223456789073	1123456789091
2323456789078	3423456789052
2923456789094	1123456789050
3323456789011	0023456789071
3323456789098	0123456789036
3423456789052	2323456789078
4423456789008	5523456789095
4423456789070	7723456789010
5523456789013	6623456789005
5523456789090	7723456789056
5523456789095	4423456789008
5623456789018	6723456789000
5901234567895	2000987654324
5906543210984	8711234567893
5909876543210	1001234567898
6623456789005	5523456789013
6623456789037	9923456781462

Рис.Б.11. Данные таблицы «Аналог»

Код программы

1. Форма «frmAuth»

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace Pharmacy
{
    public partial class frmAuth : Form
    {
        byte EnterTry = 2;

        public frmAuth()
        {
            InitializeComponent();
        }

        private void btnEntry_Click(object sender, EventArgs e)
        {
            if (EnterTry == 0) {
                btnEntry.Enabled = false;
                MessageBox.Show("У вас закончились попытки входа", "Ошибка",
                    MessageBoxButtons.OK, MessageBoxIcon.Error);
            }

            try
            {
                sqlCommand_auth.Parameters["@fio"].Value = txtLogin.Text;
                sqlCommand_auth.Parameters["@phoneNum"].Value = txtPasswd.Text;

                sqlConnection.Open();
                sqlCommand_auth.ExecuteNonQuery();
                sqlConnection.Close();
            }
            catch { }
        }
    }
}
```



```

        if (Convert.ToInt32(sqlCommand_auth.Parameters["@a"].Value) == 0)
        {
            MessageBox.Show($"Повторите ввод, количество попыток:
{EnterTry}",
            "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
            EnterTry--;
            // txtLogin.Clear();
            txtPasswd.Clear();
        }

        else if (Convert.ToInt32(sqlCommand_auth.Parameters["@a"].Value)
== 1) {
            string word = txtLogin.Text; // Получение введенного слова из
txtLogin
            Form frm = new frmCashier(word);
            frm.Show();
            this.Hide();
        }

        else if (Convert.ToInt32(sqlCommand_auth.Parameters["@a"].Value)
== 2)
        {
            string word = txtLogin.Text; // Получение введенного слова из
txtLogin
            Form frm = new frmManager(word);
            frm.Show();
            this.Hide();
        }
        else if (Convert.ToInt32(sqlCommand_auth.Parameters["@a"].Value)
== 3)
        {
            string word = txtLogin.Text; // Получение введенного слова из
txtLogin
            Form frm = new frmDirector(word);
            frm.Show();
            this.Hide();
        }
    }
    catch
    {
        MessageBox.Show($"Повторите ввод, количество попыток:
{EnterTry}", "Ошибка",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    }

```

```

        EnterTry--;
        // txtLogin.Clear();
        txtPasswd.Clear();
        sqlCommand_auth.Parameters["@fio"].Value = 0;
        sqlCommand_auth.Parameters["@phoneNum"].Value = 0;
        sqlCommand_auth.Parameters["@a"].Value = 0;
        sqlConnection.Close();
    }
}

private void txtPasswd_TextChanged(object sender, EventArgs e)
{
    txtPasswd.PasswordChar = '*'; // Установка символа замены для
    скрывания текста
}
}
}

```

2. Форма «frmCashier»

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Reflection.Emit;
using System.Security.Cryptography;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using static Pharmacy.PharmacyDataSet;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace Pharmacy
{
    public partial class frmCashier : Form
    {
        private string word; // Переменная, в которой будет храниться переданное
        слово

        public frmCashier(string word)
        {
            InitializeComponent();

```

```

        this.word = word; // Сохранение переданного слова в переменную word
    }

    private void frmCashier_Load(object sender, EventArgs e)
    {
        // TODO: данная строка кода позволяет загрузить данные в таблицу
        "pharmacyDataSet.getCashierSotr". При необходимости она может быть
        перемещена или удалена.

        this.getCashierSotrTableAdapter.Fill(this.pharmacyDataSet.getCashierSotr);
        // TODO: данная строка кода позволяет загрузить данные в таблицу
        "pharmacyDataSet._tableProduct". При необходимости она может быть
        перемещена или удалена.
        this.tableProductTableAdapter.Fill(this.pharmacyDataSet._tableProduct);
        // TODO: данная строка кода позволяет загрузить данные в таблицу
        "pharmacyDataSet.Sotr". При необходимости она может быть перемещена или
        удалена.
        this.sotrTableAdapter.Fill(this.pharmacyDataSet.Sotr);
        // TODO: данная строка кода позволяет загрузить данные в таблицу
        "pharmacyDataSet.Product". При необходимости она может быть перемещена
        или удалена.
        this.productTableAdapter.Fill(this.pharmacyDataSet.Product);
        // TODO: данная строка кода позволяет загрузить данные в таблицу
        "pharmacyDataSet.Product". При необходимости она может быть перемещена
        или удалена.
        this.productTableAdapter.Fill(this.pharmacyDataSet.Product);
        // TODO: данная строка кода позволяет загрузить данные в таблицу
        "pharmacyDataSet.Product". При необходимости она может быть перемещена
        или удалена.
        this.productTableAdapter.Fill(this.pharmacyDataSet.Product);

        txtOutput.Text = word; // Отображение переданного слова в txtOutput
    }

    private void ClearInsert(object sender, EventArgs e)
    {
        txtFind.Text = "";
    }

    private void btnReset_Click(object sender, EventArgs e)
    {
        tableProductBindingSource.Filter = "";
        ClearInsert(sender, e);
    }

```

```

private void btnFindAnalog_Click(object sender, EventArgs e)
{
    Form frm = new frmAnalog();
    frm.Show();
}

private void btnMedicament_Click(object sender, EventArgs e)
{
    Form frm = new frmMedicament();
    frm.Show();
}

private void btnBack_Click(object sender, EventArgs e)
{
    Form frm = new frmAuth();
    frm.Show();
    this.Hide();
}

private void ClearInsertSell(object sender, EventArgs e)
{
    txtBarcodeSell.Text = "";
    numUD_count.Value = 1;
}

private void RefreshProductTable()
{
    tblProduct.DataSource = null; // Очищаем источник данных
    SqlDataAdapter adapter = new SqlDataAdapter("SELECT * FROM
tableProduct", sqlConnection);
    DataTable table = new DataTable();
    adapter.Fill(table);
    BindingSource bs = new BindingSource();
    bs.DataSource = table;
    tblProduct.DataSource = bs;

    // Повторное применение фильтра после обновления таблицы
    tableProductBindingSource = bs;
    tableProductBindingSource.Filter = "NameProduct Like '" + "%" +
txtFind.Text
    + "%' OR Brand Like '%" + "%" + txtFind.Text + "%"
    + "%' OR NameCategory Like '%" + "%" + txtFind.Text + "%'";
}

private void btnSell_Click_1(object sender, EventArgs e)

```

```

    {
        if (txtBarcodeSell.Text.Length != 13)
        {
            MessageBox.Show("Введен некорректный штрихкод", "Оформление
продажи",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
            return;
        }
        foreach (char c in txtBarcodeSell.Text)
        {
            if (!char.IsDigit(c))
            {
                MessageBox.Show("Введен некорректный штрихкод",
"Оформление продажи",
                    MessageBoxButtons.OK, MessageBoxIcon.Error);
                return;
            }
        }
    }

    //проверка того, что в таблице Продукт такой товар существует
    sqlCommand_checkBarcode.Parameters["@barcode"].Value =
txtBarcodeSell.Text;

    sqlConnection.Open();
    int count = (int)sqlCommand_checkBarcode.ExecuteScalar();
    sqlConnection.Close();

    if (count == 0)
    {
        MessageBox.Show("Такого товара не существует", "Оформление
продажи",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
        // txtBarcode.Text = string.Empty;
        return;
    }

    sqlCommand_checkStock.Parameters["@barcode"].Value =
txtBarcodeSell.Text;
    sqlConnection.Open();
    int availableQuantity = (int)sqlCommand_checkStock.ExecuteScalar();
    sqlConnection.Close();

    if (availableQuantity < Convert.ToInt32(numUD_count.Text))
    {

```

```

        MessageBox.Show("Выбранное количество товара больше, чем есть
в наличии", "Оформление продажи",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }

    sqlCommand_Sell.Parameters["@barcode"].Value = txtBarcodeSell.Text;
    sqlCommand_Sell.Parameters["@quantity"].Value = numUD_count.Text;
    sqlCommand_Sell.Parameters["@sotr"].Value = txtOutput.Text;

    sqlConnection.Open();
    sqlCommand_Sell.ExecuteNonQuery();
    sqlConnection.Close();

    MessageBox.Show("Продажа оформлена", "Оформление продажи",
    MessageBoxButtons.OK, MessageBoxIcon.Information);

    ClearInsertSell(sender, e);

    RefreshProductTable();
}

private void btnFind_Click(object sender, EventArgs e)
{
    tableProductBindingSource.Filter = "NameProduct Like '" + "%" +
txtFind.Text
    + "%' OR Brand Like '%" + "%" + txtFind.Text + "%"
    + "%' OR NameCategory Like '%" + "%" + txtFind.Text + "%'";
}
}
}

```

3. Форма «frmMedicament»

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Xml.Linq;

```

```

namespace Pharmacy
{
    public partial class frmMedicament : Form
    {
        public frmMedicament()
        {
            InitializeComponent();
        }

        private void btnMedicament_Click(object sender, EventArgs e)
        {
            if (txtBarcode.Text.Length != 13)
            {
                MessageBox.Show("Введен некорректный штрихкод", "Добавление характеристик",
                    MessageBoxButtons.OK, MessageBoxIcon.Error);
                return;
            }
            foreach (char c in txtBarcode.Text)
            {
                if (!char.IsDigit(c))
                {
                    MessageBox.Show("Введен некорректный штрихкод", "Добавление характеристик",
                        MessageBoxButtons.OK, MessageBoxIcon.Error);
                    break;
                }
            }

            //проверка того, что в таблице Продукт такой товар существует
            sqlCommand_checkBarcode.Parameters["@barcode"].Value =
txtBarcode.Text;

            sqlConnection.Open();
            int count = (int)sqlCommand_checkBarcode.ExecuteScalar();
            sqlConnection.Close();

            if (count == 0)
            {
                MessageBox.Show("Такого товара не существует", "Просмотр характеристик",
                    MessageBoxButtons.OK, MessageBoxIcon.Error);
                return;
            }
        }
    }
}

```

```

        sqlCommand_medicament.Parameters["@barcode"].Value =
txtBarcode.Text;

        sqlConnection.Open();
        DataTable dt = new DataTable();
        SqlDataAdapter da = new SqlDataAdapter(sqlCommand_medicament);
        da.Fill(dt);

        if (dt.Rows.Count == 0)
        {
            MessageBox.Show("Характеристики для данного товара
отсутствуют",
                "Просмотр характеристик", MessageBoxButtons.OK,
                MessageBoxIcon.Information);
            sqlConnection.Close();
            return;
        }

        else
        {

            txtPharmGroup.Text = dt.Rows[0]["Фарм.группа"].ToString();
            txtExp.Text = dt.Rows[0]["Срок годности"].ToString();
            txtNotes.Text = dt.Rows[0]["Примечания"].ToString();
            txtEffect.Text = dt.Rows[0]["Эффект"].ToString();
            txtActiveSubstance.Text = dt.Rows[0]["Действующее
вещество"].ToString();
            txtReleaseForm.Text = dt.Rows[0]["Форма выпуска"].ToString();
            txtDosage.Text = dt.Rows[0]["Дозировка"].ToString();
            txtQuantity.Text = dt.Rows[0]["Количество"].ToString();
            txtVolume.Text = dt.Rows[0]["Объем"].ToString();
            txtComposition.Text = dt.Rows[0]["Состав"].ToString();
            txtCountry.Text = dt.Rows[0]["Страна"].ToString();
            txtManufactur.Text = dt.Rows[0]["Производитель"].ToString();

            sqlConnection.Close();

            sqlCommand_getProductInfo.Parameters["@barcode"].Value =
txtBarcode.Text;

            sqlConnection.Open();
            txtProductInfo.Text =
sqlCommand_getProductInfo.ExecuteScalar().ToString();
            sqlConnection.Close();

```



```

    }
  }
}

```

4. Форма «frmAnalog»

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace Pharmacy
{
    public partial class frmAnalog : Form
    {
        public frmAnalog()
        {
            InitializeComponent();

            private void btnAnalog_Click(object sender, EventArgs e)
            {
                if (txtBarcode.Text.Length != 13)
                {
                    MessageBox.Show("Введен некорректный штрихкод", "Просмотр аналогов",
                        MessageBoxButtons.OK, MessageBoxIcon.Error);
                    // txtBarcode.Text = string.Empty; // Очистить поле ввода
                    return;
                }
                foreach (char c in txtBarcode.Text)
                {
                    if (!char.IsDigit(c))
                    {
                        MessageBox.Show("Введен некорректный штрихкод", "Просмотр аналогов",

```

```

        MessageBoxButtons.OK, MessageBoxIcon.Error);
    //    txtBarcode.Text = string.Empty;
        break;
    }
}

//проверка того, что в таблице Продукт такой товар существует
sqlCommand_checkBarcode.Parameters["@barcode"].Value =
txtBarcode.Text;

sqlConnection.Open();
int count = (int)sqlCommand_checkBarcode.ExecuteScalar();
sqlConnection.Close();

if (count == 0)
{
    MessageBox.Show("Такого товара не существует", "Просмотр
аналогов",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    //    txtBarcode.Text = string.Empty;
    return;
}

sqlCommand_findAnalogues.Parameters["@Barcode"].Value =
txtBarcode.Text;

sqlConnection.Open();
var temp = new DataTable();
temp.Load(sqlCommand_findAnalogues.ExecuteReader());
tblAnalogues.DataSource = temp;
sqlConnection.Close();

sqlCommand_getProductInfo.Parameters["@barcode"].Value =
txtBarcode.Text;

sqlConnection.Open();
txtProductInfo.Text =
sqlCommand_getProductInfo.ExecuteScalar().ToString();
sqlConnection.Close();
}
}
}

```

5. Форма «frmManager»

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Pharmacy
{
    public partial class frmManager : Form
    {
        private string word; // Переменная, в которой будет храниться переданное
        СЛОВО

        public frmManager(string word)
        {
            InitializeComponent();
            this.word = word; // Сохранение переданного слова в переменную word
        }

        private void btnProduct_Click(object sender, EventArgs e)
        {
            Form frm = new frmProduct(word);
            frm.Show();
            this.Hide();
        }

        private void btnSotr_Click(object sender, EventArgs e)
        {
            Form frm = new frmSotr(word);
            frm.Show();
            this.Hide();
        }

        private void btnBack_Click(object sender, EventArgs e)
        {
            Form frm = new frmAuth();
            frm.Show();
            this.Hide();
        }
    }
}

```

6. Форма «frmProduct»

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Pharmacy
{
    public partial class frmProduct : Form
    {
        private string word; // Переменная, в которой будет храниться переданное
        СЛОВО

        public frmProduct(string word)
        {
            InitializeComponent();
            this.word = word; // Сохранение переданного слова в переменную word
        }

        private void frmProduct_Load(object sender, EventArgs e)
        {
            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "pharmacyDataSet1.getPurveyor". При необходимости она может быть
            перемещена или удалена.
            this.getPurveyorTableAdapter.Fill(this.pharmacyDataSet1.getPurveyor);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "pharmacyDataSet1.viewGetManagerSotr". При необходимости она может быть
            перемещена или удалена.

            this.viewGetManagerSotrTableAdapter.Fill(this.pharmacyDataSet1.viewGetMana
            gerSotr);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "pharmacyDataSet1._tableProduct". При необходимости она может быть
            перемещена или удалена.
            this.tableProductTableAdapter.Fill(this.pharmacyDataSet1._tableProduct);
```

// TODO: данная строка кода позволяет загрузить данные в таблицу "pharmacyDataSet.Product". При необходимости она может быть перемещена или удалена.

```
        this.productTableAdapter.Fill(this.pharmacyDataSet.Product);

        txtOutput.Text = word; // Отображение переданного слова в txtOutput
    }

    private void btnAddProduct_Click(object sender, EventArgs e)
    {
        Form frm = new frmAddProduct(word);
        frm.Show();
        this.Hide();
    }

    private void btnFind_Click(object sender, EventArgs e)
    {
        tableProductBindingSource.Filter = "Barcode Like '%" + txtFind.Text
        + "%' OR NameProduct Like '%" + txtFind.Text + "%'"
        + " OR Brand Like '%" + txtFind.Text + "%'"
        + " OR NameCategory Like '%" + txtFind.Text + "%'";
    }

    private void ClearInsert(object sender, EventArgs e)
    {
        txtFind.Text = "";
    }

    private void btnReset_Click(object sender, EventArgs e)
    {
        tableProductBindingSource.Filter = "";
        ClearInsert(sender, e);
    }

    private void btnBack_Click(object sender, EventArgs e)
    {
        Form frm = new frmManager(word);
        frm.Show();
        this.Hide();
    }

    private void RefreshProductTable()
    {
        tblProduct.DataSource = null; // Очищаем источник данных
```

```

        SqlDataAdapter adapter = new SqlDataAdapter("SELECT * FROM
tableProduct", sqlConnection);
        DataTable table = new DataTable();
        adapter.Fill(table);
        BindingSource bs = new BindingSource();
        bs.DataSource = table;
        tblProduct.DataSource = bs;

        // Повторное применение фильтра после обновления таблицы
        tableProductBindingSource = bs;
        tableProductBindingSource.Filter = "NameProduct Like '" + "%" +
txtFind.Text
        + "%' OR Brand Like '%" + "%" + txtFind.Text + "%"
        + "%' OR NameCategory Like '%" + "%" + txtFind.Text + "%'";
    }

    private void ClearInsertPur(object sender, EventArgs e)
    {
        txtBarcodePur.Text = "";
        numUD_count.Value = 1;
    }

    private void btnPurchase_Click(object sender, EventArgs e)
    {
        if (txtBarcodePur.Text.Length != 13)
        {
            MessageBox.Show("Введен некорректный штрихкод", "Оформление
продажи",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
            // txtBarcode.Text = string.Empty; // Очистить поле ввода
            return;
        }
        foreach (char c in txtBarcodePur.Text)
        {
            if (!char.IsDigit(c))
            {
                MessageBox.Show("Введен некорректный штрихкод",
"Оформление продажи",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
                // txtBarcode.Text = string.Empty;
                break;
            }
        }
    }

```

```

        sqlCommand_Purchase.Parameters["@barcode"].Value =
txtBarcodePur.Text;
        sqlCommand_Purchase.Parameters["@quantity"].Value =
numUD_count.Text;
        sqlCommand_Purchase.Parameters["@sotr"].Value = txtOutput.Text;
        sqlCommand_Purchase.Parameters["@purveyor"].Value =
cbPurveyor.Text;

```

```

        sqlConnection.Open();
        sqlCommand_Purchase.ExecuteNonQuery();
        sqlConnection.Close();

```

```

        MessageBox.Show("Поставка оформлена", "Оформление поставки",
MessageBoxButtons.OK, MessageBoxIcon.Information);

```

```

        ClearInsertPur(sender, e);

```

```

        RefreshProductTable();

```

```

    }

```

```

private void btnWriteOff_Click(object sender, EventArgs e)
{
    Form frm = new frmWriteOff(word);
    frm.Show();
    this.Hide();
}

```

```

private void btnMedicament_Click(object sender, EventArgs e)
{
    Form frm = new frmMedicament();
    frm.Show();
}

```

```

private void btnAddPurveyor_Click(object sender, EventArgs e)
{
    Form frm = new frmAddPurveyor(word);
    frm.Show();
    this.Hide();
}

```

```

private void btnDeleteProduct_Click(object sender, EventArgs e)
{
    Form frm = new frmSeeDeleteProduct(word);
    frm.Show();
    this.Hide();
}

```

```

    }

    private void btnSeeSell_Click(object sender, EventArgs e)
    {
        Form frm = new frmSeeSell(word);
        frm.Show();
        this.Hide();
    }

    private void btnSeePurchase_Click(object sender, EventArgs e)
    {
        Form frm = new frmSeePurchase(word);
        frm.Show();
        this.Hide();
    }
}

```

7. Форма «frmAddPurveyor»

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Pharmacy
{
    public partial class frmAddPurveyor : Form
    {
        private string word; // Переменная, в которой будет храниться переданное
        СЛОВО

        public frmAddPurveyor(string word)
        {
            InitializeComponent();
            this.word = word; // Сохранение переданного слова в переменную word
        }

        private void frmAddPurveyor_Load(object sender, EventArgs e)

```



```

    {
        // TODO: данная строка кода позволяет загрузить данные в таблицу
        "pharmacyDataSet.Purveyor". При необходимости она может быть перемещена
        или удалена.
        this.purveyorTableAdapter.Fill(this.pharmacyDataSet.Purveyor);
    }

    private void btnFind_Click(object sender, EventArgs e)
    {
        purveyorBindingSource.Filter = "NamePurveyor Like '%" + txtFind.Text +
        "%' ";
    }

    private void ClearInsert(object sender, EventArgs e)
    {
        txtFind.Text = "";
    }

    private void btnReset_Click(object sender, EventArgs e)
    {
        purveyorBindingSource.Filter = "";
        ClearInsert(sender, e);
    }

    private void ClearInsert2(object sender, EventArgs e)
    {
        txtName.Text = "";
        txtPerson.Text = "";
        txtCountry.Text = "";
        txtSity.Text = "";
        txtAddress.Text = "";
        txtPhone.Text = "";
        txtPay.Text = "";
    }

    private void brnAdd_Click(object sender, EventArgs e)
    {
        if (string.IsNullOrEmpty(txtName.Text) ||
        string.IsNullOrEmpty(txtPerson.Text) ||
        string.IsNullOrEmpty(txtCountry.Text) ||
        string.IsNullOrEmpty(txtSity.Text) ||
        string.IsNullOrEmpty(txtAddress.Text) ||
        string.IsNullOrEmpty(txtPhone.Text))
        {

```

```

        MessageBox.Show("Введите данные", "Ошибка",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }

    else
    {
        sqlCommand_addPurveyor.Parameters["@namePurveyor"].Value =
txtName.Text;
        sqlCommand_addPurveyor.Parameters["@contactPerson"].Value =
txtPerson.Text;
        sqlCommand_addPurveyor.Parameters["@country"].Value =
txtCountry.Text;
        sqlCommand_addPurveyor.Parameters["@sity"].Value = txtSity.Text;
        sqlCommand_addPurveyor.Parameters["@address"].Value =
txtAddress.Text;
        sqlCommand_addPurveyor.Parameters["@phoneNum"].Value =
txtPhone.Text;
        sqlCommand_addPurveyor.Parameters["@termsPay"].Value =
txtPay.Text;

        sqlConnection.Open();
        sqlCommand_addPurveyor.ExecuteNonQuery();
        sqlConnection.Close();
        ClearInsert2(sender, e);

        MessageBox.Show("Запрос на добавление нового поставщика
выполнен",
        "Добавление поставщика", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    }

    sqlConnection.Open();
    var temp = new DataTable();
    temp.Load(sqlCommand_allPurv.ExecuteReader());
    tblPurveyor.DataSource = temp;
    sqlConnection.Close();
}

private void btnBack_Click(object sender, EventArgs e)
{
    Form frm = new frmProduct(word);
    frm.Show();
    this.Hide();
}

```

```
}  
}
```

8. Форма «frmAddProduct»

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Data.SqlClient;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
  
namespace Pharmacy  
{  
    public partial class frmAddProduct : Form  
    {  
        private string word; // Переменная, в которой будет храниться переданное  
        СЛОВО  
  
        public frmAddProduct(string word)  
        {  
            InitializeComponent();  
            this.word = word; // Сохранение переданного слова в переменную word  
        }  
  
        private void categoryBindingNavigatorSaveItem_Click(object sender,  
EventArgs e)  
        {  
            this.Validate();  
            this.categoryBindingSource.EndEdit();  
            // this.tableAdapterManager.UpdateAll(this.pharmacyDataSet);  
        }  
  
        private void categoryBindingNavigatorSaveItem_Click_1(object sender,  
EventArgs e)  
        {  
            this.Validate();  
            this.categoryBindingSource.EndEdit();  
            // this.tableAdapterManager.UpdateAll(this.pharmacyDataSet);
```

```

    }

    private void frmAddProduct_Load(object sender, EventArgs e)
    {
        // TODO: данная строка кода позволяет загрузить данные в таблицу
        "pharmacyDataSet.Product". При необходимости она может быть перемещена
        или удалена.
        this.productTableAdapter.Fill(this.pharmacyDataSet.Product);
        // TODO: данная строка кода позволяет загрузить данные в таблицу
        "pharmacyDataSet.Category". При необходимости она может быть перемещена
        или удалена.
        this.categoryTableAdapter.Fill(this.pharmacyDataSet.Category);
    }

    private void btnCreate_Click(object sender, EventArgs e)
    {
        if (txtBarcode.Text.Length != 13) {
            MessageBox.Show("Введен некорректный штрихкод", "Добавление
товара",
                MessageBoxButtons.OK, MessageBoxIcon.Information);
            return;
        }

        foreach (char c in txtBarcode.Text) {
            if (!char.IsDigit(c)) {
                MessageBox.Show("Введен некорректный штрихкод", "Добавление
товара", MessageBoxButtons.OK, MessageBoxIcon.Information);
                // txtBarcode.Text = string.Empty; // Очистить поле ввода
                break;
            }
        }

        foreach (char c in txtPrice.Text) {
            if (!char.IsDigit(c)) {
                MessageBox.Show("Введена некорректная цена", "Добавление
товара", MessageBoxButtons.OK, MessageBoxIcon.Information);
                txtPrice.Text = string.Empty; // Очистить поле ввода
                break;
            }
        }

        if (string.IsNullOrEmpty(txtBarcode.Text) ||
            string.IsNullOrEmpty(cbCategory.Text))

```

```

        || string.IsNullOrEmpty(txtBrand.Text) ||
string.IsNullOrEmpty(txtNameProduct.Text)
        || string.IsNullOrEmpty(txtPrice.Text))
    {
        MessageBox.Show("Введены не все требуемые данные", "Ошибка",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }
    else {
        // присвоить значения входным параметрам процедуры с
преобразованием типов
        sqlCommand_addProduct.Parameters["@barcode"].Value =
txtBarcode.Text;
        sqlCommand_addProduct.Parameters["@nameCategory"].Value =
cbCategory.Text;
        sqlCommand_addProduct.Parameters["@brand"].Value = txtBrand.Text;
        sqlCommand_addProduct.Parameters["@nameProduct"].Value =
txtNameProduct.Text;
        sqlCommand_addProduct.Parameters["@price"].Value = txtPrice.Text;
        sqlCommand_addProduct.Parameters["@minReserve"].Value =
numUDMinReserve.Text;
        sqlCommand_addProduct.Parameters["@inStock"].Value =
numUDInStock.Text;
        // открыть соединение с БД
        sqlConnection.Open();
        // выполнить sql-выражение (хранимую процедуру) и вернуть
количество измененных записей
        sqlCommand_addProduct.ExecuteNonQuery();
        // закрыть соединение с БД
        sqlConnection.Close();

        // вывод результата через окно сообщения
        MessageBox.Show("Товар создан", "Создание товара",
MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

private void btnAddMedicament_Click(object sender, EventArgs e)
{
    Form frm = new frmAddMedicament();
    frm.Show();
}

private void btnBack_Click(object sender, EventArgs e)

```

```

    {
        Form frm = new frmProduct(word);
        frm.Show();
        this.Hide();
    }

    private void ClearInsert(object sender, EventArgs e)
    {
        txtBarcode.Text = "";
        cbCategory.Text = "";
        txtBrand.Text = "";
        txtNameProduct.Text = "";
        txtPrice.Text = "";
        numUDMinReserve.Value = 0;
        numUDInStock.Value = 0;
    }

    private void btnClear_Click(object sender, EventArgs e)
    {
        ClearInsert(sender, e);
    }
}

```

9. Форма «frmAddMedicament»

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Pharmacy
{
    public partial class frmAddMedicament : Form
    {
        public frmAddMedicament()
        {
            InitializeComponent();
        }
    }
}

```

```

private void btnAdd_Click(object sender, EventArgs e)
{
    // проверка уникальности штрихкода
    string query = "SELECT COUNT(*) FROM Medicament WHERE
Barcode = @barcode";

    SqlCommand command = new SqlCommand(query, sqlConnection);
    command.Parameters.AddWithValue("@barcode", txtBarcode.Text);

    sqlConnection.Open();
    int kolvo = (int)command.ExecuteScalar(); // Получаем количество
записей с введенным штрихкодом
    sqlConnection.Close();

    // Проверяем, было ли найдено совпадение
    if (kolvo > 0)
    {
        MessageBox.Show("Описание этого товара уже имеется.",
"Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }

    if (txtBarcode.Text.Length != 13)
    {
        MessageBox.Show("Введен некорректный штрихкод", "Добавление
товара", MessageBoxButtons.OK, MessageBoxIcon.Information);
        // txtBarcode.Text = string.Empty; // Очистить поле ввода
        return;
    }
    foreach (char c in txtBarcode.Text)
    {
        if (!char.IsDigit(c))
        {
            MessageBox.Show("Введен некорректный штрихкод", "Добавление
товара", MessageBoxButtons.OK, MessageBoxIcon.Information);
            // txtBarcode.Text = string.Empty; // Очистить поле ввода
            return;
        }
    }

    if (string.IsNullOrEmpty(txtBarcode.Text) ||
string.IsNullOrEmpty(txtPharmGroup.Text)
        || string.IsNullOrEmpty(cbExp.Text) ||
string.IsNullOrEmpty(txtActiveSubstance.Text)

```

```

        || string.IsNullOrEmpty(txtReleaseForm.Text) ||
string.IsNullOrEmpty(txtComposition.Text)
        || string.IsNullOrEmpty(txtManufactur.Text))
    {
        MessageBox.Show("Введены не все требуемые данные " +
            "(проверьте заполнение следующих полей: Штрихкод,
Фарм.группа, Срок годности, Действующее вещество, Форма выпуска,
Состав, Производитель)",
            "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }
    else
    {
        //проверка того, что в таблице Продукт такой товар существует
        sqlCommand_checkBarcode.Parameters["@barcode"].Value =
txtBarcode.Text;

        sqlConnection.Open();
        int count = (int)sqlCommand_checkBarcode.ExecuteScalar();
        sqlConnection.Close();

        if (count == 0)
        {
            MessageBox.Show("Такого товара не существует", "Добавление
характеристик", MessageBoxButtons.OK, MessageBoxIcon.Information);
            // txtBarcode.Text = string.Empty;
            return;
        }

        else
        {
            // присвоить значения входным параметрам процедуры с
преобразованием типов
            sqlCommand_addMedicament.Parameters["@barcode"].Value =
txtBarcode.Text;
            sqlCommand_addMedicament.Parameters["@pharmGroup"].Value =
txtPharmGroup.Text;
            sqlCommand_addMedicament.Parameters["@exp"].Value =
cbExp.Text;
            sqlCommand_addMedicament.Parameters["@notes"].Value =
txtNotes.Text;
            sqlCommand_addMedicament.Parameters["@effect"].Value =
txtEffect.Text;

```



```

sqlCommand_addMedicament.Parameters["@activeSubstance"].Value =
txtActiveSubstance.Text;
        sqlCommand_addMedicament.Parameters["@releaseForm"].Value =
txtRealeseForm.Text;
        sqlCommand_addMedicament.Parameters["@dosage"].Value =
txtDosage.Text;
        sqlCommand_addMedicament.Parameters["@quantity"].Value =
numUD_count.Text;
        sqlCommand_addMedicament.Parameters["@volume"].Value =
txtVolune.Text;
        sqlCommand_addMedicament.Parameters["@composition"].Value =
txtComposition.Text;
        sqlCommand_addMedicament.Parameters["@country"].Value =
txtCountry.Text;
        sqlCommand_addMedicament.Parameters["@manufactur"].Value =
txtManufactur.Text;
        // открыть соединение с БД
        sqlConnection.Open();
        // выполнить sql-выражение (хранимую процедуру) и вернуть
количество измененных записей
        sqlCommand_addMedicament.ExecuteNonQuery();
        // закрыть соединение с БД
        sqlConnection.Close();

        // вывод результата через окно сообщения
        MessageBox.Show("Характеристики к товару добавлены",
"Создание описания товара",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

private void ClearInsert(object sender, EventArgs e)
{
    txtBarcode.Text = "";
    txtPharmGroup.Text = "";
    cbExp.Text = "";
    txtNotes.Text = "";
    txtEffect.Text = "";
    txtActiveSubstance.Text = "";
    txtRealeseForm.Text = "";
    txtDosage.Text = "";
    numUD_count.Value = 0;
    txtVolune.Text = "";

```

```

        txtComposition.Text = "";
        txtCountry.Text = "";
        txtManufactur.Text = "";
    }

    private void btnClear_Click(object sender, EventArgs e)
    {
        ClearInsert(sender, e);
    }
}

```

10. Форма «frmWriteOff»

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace Pharmacy
{
    public partial class frmWriteOff : Form
    {
        private string word; // Переменная, в которой будет храниться переданное
        фиио

        public frmWriteOff(string word)
        {
            InitializeComponent();
            this.word = word; // Сохранение переданного фиио в переменную word
        }

        private void btnWriteOff_Click(object sender, EventArgs e)
        {
            if (txtBarcode.Text.Length != 13)
            {
                MessageBox.Show("Введен некорректный штрихкод", "Списание
                товара", MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
        }
    }
}

```

```

        txtBarcode.Text = string.Empty; // Очистить поле ввода
        return;
    }

    foreach (char c in txtBarcode.Text)
    {
        if (!char.IsDigit(c))
        {
            MessageBox.Show("Введен некорректный штрихкод", "Списание
товара", MessageBoxButtons.OK, MessageBoxIcon.Information);
            txtBarcode.Text = string.Empty;
            return;
        }
    }

    sqlCommand_checkBarcode.Parameters["@barcode"].Value =
txtBarcode.Text;

    sqlConnection.Open();
    int count = (int)sqlCommand_checkBarcode.ExecuteScalar();
    sqlConnection.Close();

    if (count == 0)
    {
        MessageBox.Show("Такого товара не существует", "Списание
товара",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }

    else
    {
        sqlCommand_writeOff.Parameters["@Barcode"].Value =
txtBarcode.Text;
        sqlCommand_writeOff.Parameters["@Quantity"].Value =
numUD_count.Text;
        sqlCommand_writeOff.Parameters["@WhyDelete"].Value =
txtWhy.Text;

        sqlConnection.Open();
        sqlCommand_writeOff.ExecuteNonQuery();
        sqlConnection.Close();

        sqlCommand_getProductInfo.Parameters["@barcode"].Value =
txtBarcode.Text;

```

```

        sqlConnection.Open();
        txtProductInfo.Text =
sqlCommand_getProductInfo.ExecuteScalar().ToString();
        sqlConnection.Close();

        MessageBox.Show("Товар списан", "Списание товара",
MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

private void frmWriteOff_Load(object sender, EventArgs e)
{
    txtWhy.Text = "Истекает срок годности";
}

private void btnBack_Click(object sender, EventArgs e)
{
    Form frm = new frmProduct(word);
    frm.Show();
    this.Hide();
}

private void btnDeleteProduct_Click(object sender, EventArgs e)
{
    Form frm = new frmSeeDeleteProduct(word);
    frm.Show();
    this.Hide();
}
}
}

```

11. Форма «frmSeeDeleteProduct»

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

```

```

namespace Pharmacy

```

```

{
    public partial class frmSeeDeleteProduct : Form
    {
        private string word; // Переменная, в которой будет храниться переданное
        фюо

        public frmSeeDeleteProduct(string word)
        {
            InitializeComponent();
            this.word = word; // Сохранение переданного фюо в переменную word
        }

        private void frmDeleteProduct_Load(object sender, EventArgs e)
        {
            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "pharmacyDataSet.deleteProductForWF". При необходимости она может быть
            перемещена или удалена.

            this.deleteProductForWFTableAdapter.Fill(this.pharmacyDataSet.deleteProductFo
            rWF);

        }

        private void btnFind_Click(object sender, EventArgs e)
        {
            deleteProductForWFBindingSource.Filter = "Barcode Like '" + "%" +
            txtFind.Text
            + "%' OR Brand Like '%" + "%" + txtFind.Text + "%"
            + "%' OR NameProduct Like '%" + "%" + txtFind.Text + "%'";
        }

        private void ClearInsert(object sender, EventArgs e)
        {
            txtFind.Text = "";
        }

        private void btnReset_Click(object sender, EventArgs e)
        {
            deleteProductForWFBindingSource.Filter = "";
            ClearInsert(sender, e);
        }

        private void btnBack_Click(object sender, EventArgs e)
        {
            Form frm = new frmProduct(word);

```

```

        frm.Show();
        this.Hide();
    }
}

```

12. Форма «frmSeeSell»

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Pharmacy
{
    public partial class frmSeeSell : Form
    {
        private string word; // Переменная, в которой будет храниться переданное
        фиио

        public frmSeeSell(string word)
        {
            this.word = word; // Сохранение переданного фиио в переменную word
            InitializeComponent();
        }

        private void btnSeeSell_Load(object sender, EventArgs e)
        {
            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "pharmacyDataSet.viewSell". При необходимости она может быть перемещена
            или удалена.
            this.viewSellTableAdapter.Fill(this.pharmacyDataSet.viewSell);
        }

        private void btnFind_Click(object sender, EventArgs e)
        {
            viewSellBindingSource.Filter = "Barcode Like '" + "%" + txtFind.Text
            + "%' OR Brand Like '%" + "%" + txtFind.Text + "%"
            + "%' OR NameProduct Like '%" + "%" + txtFind.Text + "%'";
        }
    }
}

```

```

private void ClearInsert(object sender, EventArgs e)
{
    txtFind.Text = "";
}

private void btnReset_Click(object sender, EventArgs e)
{
    viewSellBindingSource.Filter = "";
    ClearInsert(sender, e);
}

private void btnBack_Click(object sender, EventArgs e)
{
    Form frm = new frmProduct(word);
    frm.Show();
    this.Hide();
}
}
}

```

13. Форма «frmSeePurchase»

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Pharmacy
{
    public partial class frmSeePurchase : Form
    {
        private string word; // Переменная, в которой будет храниться переданное
        фиио

        public frmSeePurchase(string word)
        {
            InitializeComponent();
            this.word = word; // Сохранение переданного фиио в переменную word
        }
    }
}

```

```

private void frmSeePurchase_Load(object sender, EventArgs e)
{
    // TODO: данная строка кода позволяет загрузить данные в таблицу
    "pharmacyDataSet1.viewPurchase". При необходимости она может быть
    перемещена или удалена.

    this.viewPurchaseTableAdapter.Fill(this.pharmacyDataSet1.viewPurchase);
    // TODO: данная строка кода позволяет загрузить данные в таблицу
    "pharmacyDataSet.viewPurchase". При необходимости она может быть
    перемещена или удалена.
    this.viewPurchaseTableAdapter.Fill(this.pharmacyDataSet.viewPurchase);

}

private void btnBack_Click(object sender, EventArgs e)
{
    Form frm = new frmProduct(word);
    frm.Show();
    this.Hide();
}

private void btnFind_Click(object sender, EventArgs e)
{
    viewPurchaseBindingSource1.Filter = "NamePurveyor LIKE '%" +
txtFind.Text + "%' " +
    "OR Fio LIKE '%" + txtFind.Text + "%' " +
    "OR Barcode LIKE '%" + txtFind.Text + "%' " +
    "OR Brand LIKE '%" + txtFind.Text + "%' " +
    "OR NameProduct LIKE '%" + txtFind.Text + "%'";
}

private void ClearInsert(object sender, EventArgs e)
{
    txtFind.Text = "";
}

private void btnReset_Click(object sender, EventArgs e)
{
    viewPurchaseBindingSource1.Filter = "";
    ClearInsert(sender, e);
}
}
}

```


14. Форма «frmSotr»

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace Pharmacy
{
    public partial class frmSotr : Form
    {
        private string word; // Переменная, в которой будет храниться переданное
        СЛОВО

        public frmSotr(string word)
        {
            InitializeComponent();
            this.word = word; // Сохранение переданного слова в переменную word
        }

        private void frmSotr_Load(object sender, EventArgs e)
        {
            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "pharmacyDataSet.Sotr". При необходимости она может быть перемещена или
            удалена.
            this.sotrTableAdapter.Fill(this.pharmacyDataSet.Sotr);
        }

        private void ClearInsert(object sender, EventArgs e)
        {
            txtFio.Text = "";
            cbPosition.Text = "";
            txtEmail.Text = "";
            txtPhone.Text = "";
        }

        private void btnAddSotr_Click(object sender, EventArgs e)
        {

```

```

// проверка уникальности номера
string query = "SELECT COUNT(*) FROM Sotr WHERE PhoneNum =
@phoneNum";

SqlCommand command = new SqlCommand(query, sqlConnection);
command.Parameters.AddWithValue("@phoneNum", txtPhone.Text);

sqlConnection.Open();
int count = (int)command.ExecuteScalar(); // Получаем количество
записей с введенным номером
sqlConnection.Close();
// Проверяем, было ли найдено совпадение
if (count > 0)
{
    MessageBox.Show("Данный номер телефона уже имеется в базе
данных", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
    return;
}

if (string.IsNullOrEmpty(txtFio.Text) ||
string.IsNullOrEmpty(cbPosition.Text))
{
    MessageBox.Show("Введите данные", "Ошибка",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
    return;
}

else
{
    sqlCommand_addSotr.Parameters["@fio"].Value = txtFio.Text;
    sqlCommand_addSotr.Parameters["@position"].Value =
cbPosition.Text;
    sqlCommand_addSotr.Parameters["@email"].Value = txtEmail.Text;
    sqlCommand_addSotr.Parameters["@phoneNum"].Value =
txtPhone.Text;
    sqlConnection.Open();
    sqlCommand_addSotr.ExecuteNonQuery();
    sqlConnection.Close();
    ClearInsert(sender, e);

    MessageBox.Show("Запрос на добавление новго сотрудника
выполнен",
    "Добавление сотрудника", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
}

```

```

        sqlConnection.Open();
        var temp = new DataTable();
        temp.Load(sqlCommand_allSotr.ExecuteReader());
        tblSotr.DataSource = temp;
        sqlConnection.Close();
    }

    private void btnRemoveSotr_Click(object sender, EventArgs e)
    {
        if (tblSotr.SelectedRows.Count > 0)
        {
            int selectedIndex = tblSotr.SelectedRows[0].Index;
            int idSotr = Convert.ToInt32(tblSotr[0, selectedIndex].Value);
            sqlConnection.Open();
            using (SqlCommand sqlCommand_retireSotr
                = new SqlCommand("UPDATE Sotr SET Position = 'Уволен ' +
Position WHERE idSotr = @idSotr;", sqlConnection))
            {
                sqlCommand_retireSotr.Parameters.AddWithValue("@idSotr",
idSotr);
                sqlCommand_retireSotr.ExecuteNonQuery();
            }
            sqlConnection.Close();
        }

        else
        {
            MessageBox.Show("Выберите сотрудника", "Увольнение
сотрудника", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }

        sqlConnection.Open();
        var temp = new DataTable();
        temp.Load(sqlCommand_allSotr.ExecuteReader());
        tblSotr.DataSource = temp;
        sqlConnection.Close();
    }

    private void btnBack_Click(object sender, EventArgs e)
    {
        Form frm = new frmManager(word);
        frm.Show();
        this.Hide();
    }

```

```
    }  
}
```

15. Форма «frmDirector»

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
  
namespace Pharmacy  
{  
    public partial class frmDirector : Form  
    {  
        private string word; // Переменная, в которой будет храниться переданное  
        слово  
  
        public frmDirector(string word)  
        {  
            InitializeComponent();  
            this.word = word; // Сохранение переданного слова в переменную word  
        }  
  
        private void btnProduct_Click(object sender, EventArgs e)  
        {  
            Form frm = new frmProduct(word);  
            frm.Show();  
            this.Hide();  
        }  
  
        private void btnSotr_Click(object sender, EventArgs e)  
        {  
            Form frm = new frmSotr(word);  
            frm.Show();  
            this.Hide();  
        }  
  
        private void btnCassa_Click(object sender, EventArgs e)  
        {  
            Form frm = new frmCashier(word);
```

```

        frm.Show();
        this.Hide();
    }

    private void btnBack_Click(object sender, EventArgs e)
    {
        Form frm = new frmAuth();
        frm.Show();
        this.Hide();
    }

    private void btnSotr_Click_1(object sender, EventArgs e)
    {
        Form frm = new frmSotr(word);
        frm.Show();
        this.Hide();
    }
}

```