

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное
учреждение высшего образования
«Тольяттинский государственный университет»

Кафедра «Прикладная математика и информатика»
(наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки, специальности)

Бизнес-информатика

(направленность (профиль) / специализация)

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)**

на тему Разработка информационной системы учета лекарственных средств
на аптечном складе

Обучающийся

К. И. Сурнин

(Инициалы Фамилия)

(личная подпись)

Руководитель

канд. техн. наук Д. Г. Токарев

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2024

Аннотация

Тема «Разработка информационной системы учёта лекарственных средств на аптечном складе».

Структура настоящего проекта предусматривает наличие введения, основную часть работы, которая представлена тремя главами, а также 4 приложения. В целом, пояснительная записка имеет 60 листов.

Среди прочего, в работе имеется 37 рисунков, 12 таблиц, а также 25 литературных источников.

ПЗ, программная часть ВКР (основные и дополнительные текстовые, графические материалы), презентация – все это подается на съемном носителе, объем которого составляет 4 Гб. В графической части работы предусмотрена презентация, содержащая 12 слайдов-картинок.

Разработка приложения осуществлялась с помощью IDEVisualStudio с использованием языка программирования C#.

Содержание

Введение	4
1 Обзор путей и средств достижения цели.....	5
2 Проектирование АИС по учёту лекарственных средств	9
2.1 Информационно-логическое (Инфологическое) проектирование БД	9
2.2 Определение требований к операционной обстановке	10
2.3 Выбор СУБД и инструментальных программных средств	13
2.3.1 Характеристики производительности системы.....	14
2.3.2 Функциональные возможности, которые обеспечивают развитие ИС в перспективе	15
2.3.3 Инструментарий внутри системы для администрирования данных	16
2.3.4 Удобство и надежность Microsoft SQL Server	16
2.3.5 Доступность, ценовая политика ПО.....	17
2.4 Логическое проектирование БД	18
2.5 Физическое проектирование БД.....	23
2.6 Разработка подсистемы безопасности.....	26
3 Тестирование реализованного приложения	28
4 Оценка и обоснование экономической эффективности разработки информационной системы.....	42
Заключение.....	46
Список используемых источников	47
Приложение А.....	49
Приложение Б	50
Приложение В	57
Приложение Г	59

Введение

Тема настоящей работы в полной мере отвечает проекту, который будет реализован на практике, а именно – систематизация комплексного процесса, в рамках которого проводится учет лекарственных препаратов, что хранятся на складе аптечного предприятия.

Прежде всего, предстоит разработать базу, в которой будут храниться сведения об аптеках, лекарственных средствах, а также компаниях-производителях таких препаратов. Исключительно такой подход даст возможность в кратчайшие сроки установить, какая компания выпустила препарат; определить аптечное предприятие, в ассортименте которого можно купить препарат прямо сейчас; установить перечень аналогичных по свойствам препаратов и так далее [1].

Принимая во внимание поставленную задачу, выдвигаются некоторые требования к аптекарю:

- распределение ЛС по аптекам;
- систематизация базы согласно имеющимся в наличии лекарствам;
- внесение, а также удаление новых препаратов в базу, в том числе тех, срок годности которых истекает;
- ведение учета ЛС, продажа которых предусматривает наличие у покупателя рецепта от врача;
- изменение стоимости ЛС.

В свою очередь, покупатель:

- сможет ознакомиться в любое время с информацией, касающейся интересующего препарата;
- будет оперативно искать аптеку, в которой доступно в продаже ЛС.

Еще одна задача заключается в том, чтобы разработать справку, на основании которой можно констатировать факт наличия препаратов в конце дня. В работе ведется создание отчета, позволяющего уточнить количество проданной продукции, а также ее стоимость.

1 Обзор путей и средств достижения цели

Чтобы достичь поставленной цели, связанной с разработкой новой информационной системы (далее по тексту – ИС), допускается применять ряд некоторых средств, а также методов:

- Планирование. Необходимо оценить использование информации и информационных технологий во всей организации и саму ИС. Это можно сделать при помощи внутренних и внешних экспертов, конечных пользователей [4].

- техническое задание. Определено конечное назначение проекта, системы, в целом. Описана работа по автоматизации ИС, основные бизнес-процессы. Тут же представлены требования к информационной системе;

- Практическое применение методологии. Известно о том, что с помощью методологии можно комплексно описать процедуру, в рамках которой будет осуществлена проработка, а также сопровождение ИС. Структура описания в данном случае – жизненный цикл. Другими словами, тот или иной этап предусматривает установление системы, в которой работы идут одна за другой в последовательном порядке. На выходе стоят итоги, средства и методы, с определением уровня ответственности для каждого участника системы.

- Использование конкретных инструментов. Практика показывает, что данный подход гарантирует высокий уровень автоматизации каждого процесса, определение которого заложено в рамках методологии [7].

- развитие, а также выстраивание модельного ряда. Базой разработки информационной системы принято считать созданные, поставленные в определенной последовательности модели, которые, кроме прочего, согласуются между собой. В результате модельный ряд сохраняется, генерируется в репозитории проекта. Дополнительно приходится задействовать CASE-средства, чтобы проработать модель, трансформировать

ее по мере потребности, контролировать ее работоспособность.

В рамках данного исследования проектирование диаграмм осуществляется на основании распространенного редактора, который называется Visio.

Visio – это продукт, разработанный корпорацией Майкрософт; представляет собой графический векторный редактор. С помощью указанного инструмента можно вносить коррекцию в диаграммы, а также блок-схемы. Работает приложение на базе ОС Windows. На сегодняшний день приобрести Visio можно в одной с 3х редакций: версия стандартная, профессиональная, а также ПРО-версия. Как и в случае с AdobeReader, в Майкрософт Офис в ее стандартном исполнении, допускается возможность работы исключительно со средством визуализации диаграмм, их печати. Инструмент, который позволяет реализовать перечисленные выше задачи – Майкрософт VisioViewer [2].

Известно о том, что с самого начала анализируемый инструмент разрабатывался другой корпорацией – Shapeware, но несколько позднее она стала называться VisioCorporation. Окончательно разработка стала достоянием Майкрософт в 2000 году, когда и состоялась презентация нового продукта под названием Visio-2000. Даже если принять во внимание тот факт, что инструмент имеет непосредственную связь с корпорацией Майкрософт, в частности, в Майкрософт Офис Visio ни разу не интегрировали. Продажа программного средства была отдельной [20].

Как и во многих других проектах, которые сопряжены с достижением аналогичных целей, подобных настоящей работе, для создания базы данных применяется средство Майкрософт SQLServer. Сокращенно продукт называется так: РСУБД или система, на основании которой осуществляется управление реляционными базами данных. В основу работы данного инструментария закладывается язык запросов Transact-SQL. Принято утверждать о том, что данный язык, в свою очередь, построен на основе стандарта ANSI/ИСО, предусматривающий ряд дополнительных расширений. Эксперты применяют данное средство для реализации широкого спектра

задач: от малых БД до промышленных баз данных, которые будут интегрированы в работу крупного бизнеса [11].

Непосредственная реализация приложения протекает на основе VisualStudio. С помощью обозначенного инструментария создаются игры, консольные утилиты, а также программы, оперирующие средствами графического интерфейса.

Кроме прочего, в настоящее время VisualStudio может функционировать совместно с такими известными платформами, как .NET CompactFramework, .NET Core, MAUI и многими другими. Примечательным является тот факт, что корпорация Майкрософт, среди прочего, приобретает Xamarin. Сразу после того открылись возможности для создания программ для Android и IOS-платформы.

Что касается части работы с кодом, о в VisualStudio есть редактор, позволяющий обрабатывать исходный код. Особого внимания заслуживает технология, которая называется IntelliSense. С помощью этого средства специалист может на примитивном уровне осуществлять рефакторинговые операции с кодом. В продолжении тематики исходного кода, отметим, что есть интегрированный отладчик, который называется Майкрософт VisualStudioDebugger. Это уникальное средство, которое дает возможность применять его не только в качестве отладчика уровня, заданного внутри исходного кода, но и как отладчик, функционирующий на машинном уровне. Среди прочего, есть редактор форм, благодаря которому разрабатывается графический интерфейс для пользователя; дизайнер схематического оформления БД, а также веб-редактор. Добавим, что с помощью VisualStudio на сегодняшний день открыты возможность разработки, а также интеграции дополнительных плагинов, за счет которого можно на любом уровне расширить имеющийся функционал:

- появляется функция поддержки контрольных систем для версий, предусматривающих работу с исходным кодом;

- появляется уникальная задача для того, чтобы интегрировать новые инструменты. С их помощью выполняют операции, непосредственным образом связанные с визуальным проектированием и редактированием. Особенно это касается ситуаций, в которых будут использоваться предметно-направленные языки программирования [3];
- используются прочие инструменты для эффективной проработки ПО.

2 Проектирование АИС по учёту лекарственных средств

2.1 Информационно-логическое (Инфологическое) проектирование БД

Для дальнейшей работы предстоит сформировать список центральных объектов, которые будут представлены в разрабатываемой базе данных [5]:

1. Сущности, которые выражены такими компонентами, как ПРОИЗВОДИТЕЛЬ, АПТЕКА, ЛС (лекарственное средство).

ЛС представляет собой сущность, которая имеет следующие характеристики:

- Ссылочный атрибут, который выражается как ID препарата
- Атрибут описания, отраженный противопоказаниями к использованию ЛС;
- Еще один атрибут описательного характера – стоимость, масса, название, количество, срок годности, возможность реализации ЛС по рецепту. Теперь целесообразно представить характеристики для такой сущности, как АПТЕКА. Здесь в качестве описательных атрибутов выступает название предприятия, а также адрес, номер телефона. Есть частный компонент, как номер аптеки, что относится к указательному атрибуту.

Рассмотрим характеристики, которые представлены у сущности ИЗГОТОВИТЕЛЬ. Итак, к описательным атрибутам относятся название, адрес, телефонный номер [10].

2. Разновидности выстраиваемых связей.

- 1:М (АПТЕКА и ПРЕПАРАТ); известно о том, что в рамках одного аптечного пункта могут быть реализованы многочисленные ЛС.
- 1:М (ИЗГОТОВИТЕЛЬ - ПРЕПАРАТ). Как правило, одна компания выпускает широкую номенклатуру лекарственных препаратов, а не специализируется лишь на выпуске единичной продукции.

На основании полученной информации непосредственно можно ER-Диаграммы (Рисунок 1).

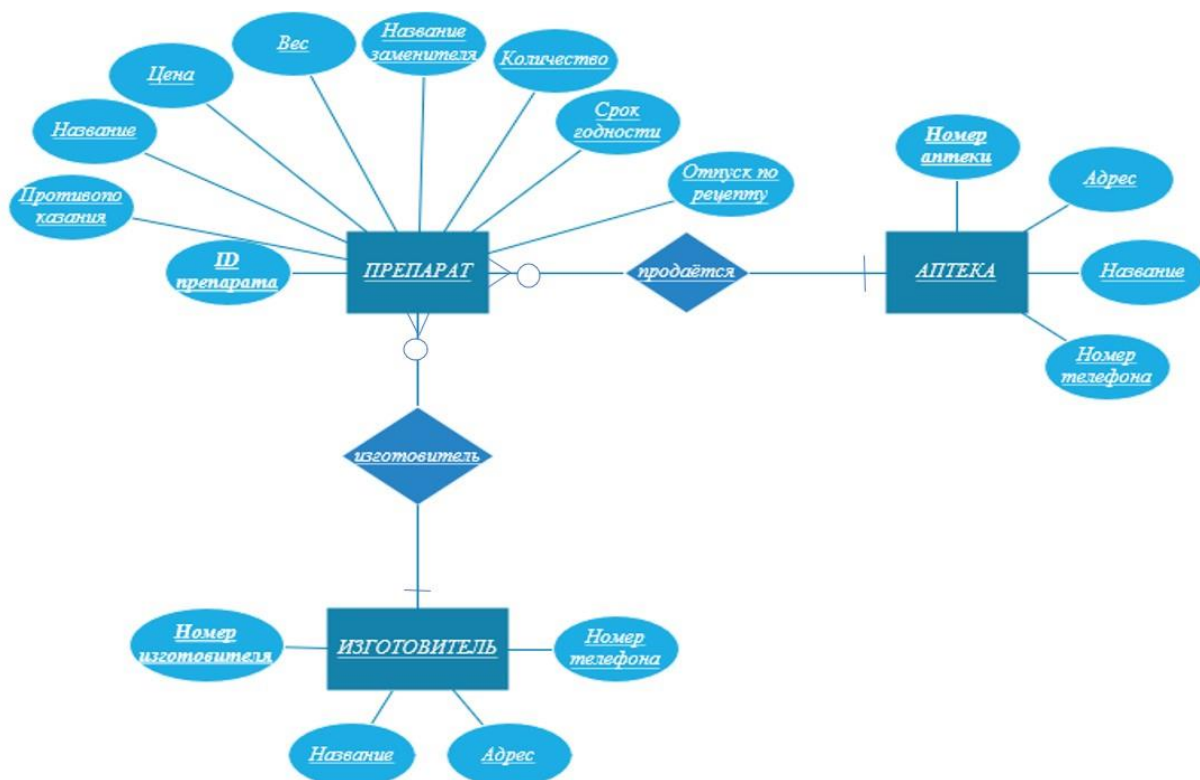


Рисунок 1 – ER – диаграмма аптеки

2.2 Определение требований к операционной обстановке

Стоит приоритетная задача – оценить комплекс требований, который будет предъявлен в отношении ресурсов для вычисления, с помощью которых обеспечивается бесперебойная работа системы [6]. В данной ситуации в целях расчета применяется формула, представленная ниже:

$$M_{\partial} = 2 \sum_{l=1} l_i \times (N_i + N_{ai})$$

Где l_i - длина записей в i -й таблице;

N_i - такое количество записей, которое по максимуму может быть представлено в i -й таблице;

N_a - численность записей, попадающих в архив i -й таблицы.

Если обратить внимание на формулу, то в ней перед знаком суммы стоит коэффициент 2. Это атрибут, который позволяет выделить память, как резерв для хранения промежуточной информации, всевозможных индексных

значений. Наконец, коэффициент помогает проводить такие объемные операции, как сортировку и многое другое [13].

Далее в рамках ВКР проводится расчет требуемых сведений для поликлиники. Чтобы достичь поставленной цели, предстоит провести расчетную операцию, а также суммирование показателей для таких сущностей, как ЛС, АПТЕКА, ИЗГОТОВИТЕЛЬ.

Для начала будет проведена работа с сущностью ЛС. Открывается Таблица 1, в рамках которой будут выделены все атрибуты, разновидности информации под них, в том числе укажем на ресурс памяти, который будет занят представленными атрибутами [9].

Таблица 1 - Сущность ЛС

<u>ID</u> <u>препарата</u>	<u>Противопо</u> <u>казания</u>	<u>Название</u>	<u>Цена</u>	<u>Вес</u>	<u>Название</u> <u>заменителя</u>	<u>Количество</u>	<u>Срок годности</u>	<u>Отпуск по</u> <u>рецепту</u>
VARCHAR (10 б.)	VARCHAR (45 б.)	VARCHAR (20 б.)	VARCHAR (6 б.)	VARCHA R (10 б.)	VARCHAR (45 б.)	INT (4 б.)	VARCHAR (10 б.)	VARCHAR (10 б.)

Допускаем такую ситуацию, согласно которой доступными оказываются 100 препаратов. Соответственно, сущность займет следующий объем памяти:

$$M_1 = 2 * 100 * 160 = 32000 \text{ байт.}$$

Теперь будет проанализирована сущность, которая называется ИЗГОТОВИТЕЛЬ. Для нее предстоит провести точно такую работу, как и по предыдущей сущности, составив Таблицу 2.

Таблица 2 - Сущность ИЗГОТОВИТЕЛЬ

<u>Номер</u> <u>изготовителя</u>	<u>Название</u>	<u>Адрес</u>	<u>Номер телефона</u>
VARCHAR (10 б.)	VARCHAR (25 б.)	VARCHAR (80 б.)	VARCHAR (11 б.)

В данной ситуации смоделируем случай, при котором рассматривается только 30 производителей. Следовательно, анализируемая сущность займет память в количестве:

$$M_2 = 2 * 30 * (10 + 25 + 80 + 11) = 7560 \text{ байт.}$$

Аналогичные операции проводятся для сущности АПТЕКА – Таблица 3.

Таблица 3 - Сущность АПТЕКА

<u>Номер аптеки</u>	<u>Название</u>	<u>Адрес</u>	<u>Номер телефона</u>
VARCHAR (10 б.)	VARCHAR (25 б.)	VARCHAR (80 б.)	VARCHAR (11 б.)

Допустим, что работает 30 аптек, под которые выделяется память в количестве:

$$M_3 = 2 * 30 * (10 + 25 + 80 + 11) = 7560 \text{ байт.}$$

Полученные данные суммируются в значении М. Это позволяет провести окончательный расчет требуемого резерва памяти:

$$M_1 + M_2 + M_3 = 7560 + 32000 + 7560 = 47120 \text{ байт.}$$

Принято считать, что в течение всего времени работы аптечного предприятия данные будут приращиваться в средних темпах. Однако динамика роста будет существенной.

Не менее значимым является вопрос, который касается характера, а также степени активности совершения запросов к БД.

Самые распространенные запросы, которые совершаются к данным – это получение записей, их удаление по мере потребности, редактирование и просто визуальное ознакомление. Что касается интенсивности самих запросов, то она средне-низкая. В остальном этот параметр определяется численностью собственников, так как регулярно возникают задачи, связанные с обновлением списка препаратов, производителей, торговых точек, прочее [14].

Проанализируем требования, которые будут предъявлены к времени, за которое система даст отклик на тот или иной запрос. Принимая во внимание тот факт, что собственники выстроены в большой поток, важно задать самое короткое время на отклик по тому или иному запросу.

То есть, практически предстоит выбрать конфигурационную структуру

ЭВМ, при которой в точности может быть достигнута поставленная цель-оперативность в плане срабатывания системы. Кроме прочего, нельзя не отметить тот факт, что новая БД должна постоянно обновляться. Самое главное, чтобы предложенный проект имел экономическую целесообразность.

Приведем конкретный пример:

Процессор - IntelCorei5 9600HQ;

ОЗУ - 16GbDDR-4 2133MHz;

Дисковая подсистема – SATA 6 Гбит/с.

С учетом того, какая имеется предметная сфера, выбираем самое подходящее средство для реализации поставленной задачи. Речь идет об ОС семейства Windows 10. Сразу стоит сделать оговорку, что 10 версия ОС требует большего количества ресурсов, чем версия 7. Тем не менее, в 10-ке можно работать с многочисленными СУБД, а также регулярно обновлять последние.

2.3 Выбор СУБД и инструментальных программных средств

Практика показывает, что в РСУБД информация, как правило, хранится в форме таблиц [11]. Сведения, которые имеют между собой определенные связи, также группируются в рамках таблиц. В перспективе между самими таблицами допускается выстраивание логических цепей связей. Таким образом, к информации на сервере пользователь будет иметь доступ через приложения. Что касается администраторов, в обязанности которых ложится функция администрирования, конфигурирования, обеспечения работоспособности БД, то они будут иметь прямой доступ к ресурсам сервера. В плане такого признака, как масштабируемость, с уверенностью можно утверждать о том, что SQL Server как раз и является масштабируемой БД. Другими словами, это средство, обеспечивающее возможность хранения неограниченно большого количества информации. Кроме прочего, доступна многопользовательская работа.

Страницы будут иметь небольшой размер, не более 8 килобайт. Следовательно, можно допускать, что их открытие займет считанные секунды. В системе есть инструментарий, позволяющий обрабатывать транзакции в

режиме онлайн. По мере потребностей можно воспользоваться функционалом, что предусматривает динамическую блокировку.

Комплекс задач, которые обычно выполняет администратор, переведены в автоматизированный режим исполнения. Прежде всего, в данной ситуации мы говорим об управлении ресурсами памяти, о блокировке, об изменении размеров файлов. Система обладает смарт-настройками, с возможностью открытия пользовательских профилей.

Чтобы найти нужную информацию за считанные секунды, доступен поиск по тексту с применением ключевых слов, фраз. Среди прочего, SQL Server предусматривает связь через Интернет и синхронизацию.

Выделим другую значимую характеристику будущей ИС. Она состоит в том, чтобы применять сервер для проведения интерактивного анализа. Это инструментарий, который дает возможность принимать взвешенные решения, создавать отчетную документацию предприятия. Чтобы создать требуемый запрос, работы проводятся на английском. Наконец, СУБД может функционировать с прочими распространенными ПП, которые предлагает корпорация Майкрософт [22].

Принимая во внимание обозначенные параметры выше, с уверенностью можно утверждать о том, что выбранная для реализации настоящего проекта СУБД под названием SQL Server идеальна для предметной сферы. Во многом это обуславливается наличием функций администрирования, расширенных функциональных возможностей, в целом. Особенно отметим качество кроссплатформенности.

2.3.1 Характеристики производительности системы

Практика показывает, что во многом популярность продукта Майкрософт SQL Server была обеспечена благодаря усилиям, которые предприняли в свое время разработчики этой корпорации. В частности, они начали поддерживать работу приложения только на базе платформы Windows NT.

Если говорить о том, почему же именно SQL Server популярна в сравнении с аналогами, отметим саму платформу Windows 2000, на которой продукт и функционирует сегодня. Не поддается сомнению значимость связи платформы с ОС. Далее можно говорить о таких преимуществах, как возможность масштабирования, высокий уровень надежности, а также производительности. В зависимости от того, какие формации Windows в перспективе будут появляться на рынке, будет определяться тем самым и распространенность, в том числе популярность SQL Server.

Данные, на основании которых функционирует ИС, могут быть оптимизированы. Для этого применяется соответствующий набор инструментов. Потом после обработки информация сохраняется в основной памяти, в буферном разделе. Тем не менее, получение доступа к информации, возможность изменения данных построены на принципе выгрузки данных в любое время с дискового пространства.

Существует масса ответственных операций, транзакций, в которых используются таблицы. В данной ситуации с применением технологии Nekaton можно непосредственно в основной памяти размещать таблицы внутри оптимизированных структур. Что касается прочих таблиц, с которыми работает приложение, то они сохраняются в привычных структурах. За счет данного подхода удастся выделить максимально доступные и оптимальные способы применения памяти; нет никакой потребности для того, чтобы вводить в работу множество механизмов. В случае накопления избыточной информации или ненужных файлов, внутри основной памяти происходит структурирование – удаление ненужной информации.

2.3.2 Функциональные возможности, которые обеспечивают развитие ИС в перспективе

По мере того, как на рынке появляется новая версия Майкрософт SQL Server, открываются новые функции, появляются возможности для оптимизации БД [12].

2.3.3 Инструментарий внутри системы для администрирования данных

Анализируемая платформа содержит большое количество средств, позволяющих вести удобное администрирования, а также разработки. Вместе с тем, имеется много инструментов, которые в том помогают:

Idera SQL check	поддерживает Microsoft SQL Server. Инструмент мониторинга, позволяющий получать базовую информацию о производительности сервера. Собирает около двадцати показателей: операции чтения/записи, кеш, транзакции, компиляция и перекомпиляция запросов, загрузка CPU и прочие. Лицензия: бесплатно
dbForge Studio for SQL Server	поддерживает Microsoft SQL Server. Среда разработки, предоставляющая решение для основных задач DBA и позволяющая без проблем работать со сложными проектами. Лицензия: от бесплатно до \$300
SQL Server Management Studio	поддерживает Microsoft SQL Server. Утилита из Microsoft SQL Server 2005 и более поздних версий для конфигурирования, управления и администрирования всех компонентов Microsoft SQL Server. Утилита включает скриптовый редактор и графическую программу, которая работает с объектами и настройками сервера. Лицензия: бесплатно
tSQLt	поддерживает Microsoft SQL Server. Фреймворк unit-тестов с открытым исходным кодом. Удобен тем, что во время разработки не придется переключаться между различными инструментами для создания кода и тестов. Лицензия: бесплатно
SQL Sentry Plan Explorer	поддерживает Microsoft SQL Server. Позволяет просматривать план выполнения различными способами и легко переключаться между многочисленными запросами, оценивать и настраивать запросы при помощи интуитивного анализа плана выполнения. Лицензия: от бесплатно до \$100

2.3.4 Удобство и надежность Microsoft SQL Server

В целом, нужно подчеркнуть, что Майкрософт SQL Server выступает в качестве прогрессивной БД, которая может быть задействована в плане достижения той или иной цели. Кроме прочего, в перспективе есть возможность для расширения продукта, без понижения скорости работы. Добавление пользователей осуществляется методом обновления имеющегося оборудования.

Аспект безопасности. Информация пользователя оказывается защищенной от НСД, так как продукт интегрирован с сервером безопасности. Принимая во внимание тот факт, что выстроена структура безопасности на уровне пользователя, каждый пользователь получает ограничение доступа при записи информации. Это решение обуславливается тем, что требуется защита от модификации данных.

После того, как БД обрабатывает пользовательский запрос, на монитор выводится обработанный результат. Через сеть передается минимальное количество информации. Таким образом, повышается время отклика, а также нивелируются так называемые «узкие места» по сетевой части. Вместе с тем, допускается применять анализируемую БД как базу данных для работы в Интернете.

2.3.5 Доступность, ценовая политика ПО

Известно о том, что лицензионная версия программного обеспечения версии 2019 года обойдется порядка 56000 рублей. Это среднерыночная цена. Однако можно воспользоваться некоммерческой версией приложения. Приведем сравнение, согласно которому Майкрософт SQL Server сравнивается с СУБД (Таблица 4).

Таблица 4 – Сравнительная таблица

Характеристики	MS SQL	PostgreSQL	MySQL
Тип модели	Реляционный	Реляционный	Реляционный
Производительность	Производительность СУБД, высокая, базируется на мощных системах для работы с базами данных. Требовательна к ресурсам	СУБД способна обрабатывать большие по объёму данные. Скорость работы может падать во время проведения пакетных операций или выполнения запросов чтения. Поддерживается асинхронная репликация	Производительность данной СУБД выше средней за счёт механизма многопоточности, дисковых таблиц на основе B-деревьев со сжатием индексов, базирующейся на потоках системы распределения памяти, оптимизированного метода однопроходного мультисоединения, кеш-таблиц
Степень оснащённости инструментарием	Имеются различные средства для создания баз данных, работы с информацией баз данных, перенесения данных из других систем и в другие системы.	Имеет такие инструменты как psql, datagrip, pgAdmin, EMS Studio, NAVICAT, PGCLL, phppadmin	MySQL содержит в себе множество средств разработки и администрирования, степень оснащённости высокая и удовлетворяет всем потребностям
Безопасность	Безопасность и соответствие основным и часто встречающимся требованиям. Поддерживается dynamicdatamasking и temporaldatasupport.	У PostgreSQL отсутствуют: Сертификация средств хранения персональных данных; Средства защиты БД от привилегированных пользователей; Шифрование данных на средствах хранения, при передаче по сети, а также в резервных копиях и выгрузках; Мониторинг и перехват опасного SQL трафика; Единое средство управление ключами доступа СУБД и шифрования данных	MySQL обеспечивает безопасность доступа пользователей к серверу с помощью идентификатора пользователя и зашифрованного пароля. Как и любой другой сервер базы данных, MySQL использует соответствующие средства защиты физического, сетевого доступа и файловой системы.
Лицензия	Проприетарная	Лицензия PostgreSQL (бесплатное ПО с открытым исходным кодом, либеральная лицензия)	GPL v2 или проприетарная

Поэтому был выбран MSSQLServer.

Поэтому был выбран MSSQLServer.

Говоря об инструментальных средствах, в рамках настоящего исследования целесообразно отдать предпочтение VisualStudio, ведь ее функционал является глобальным.

2.4 Логическое проектирование БД

На основании модели 4 НФ в рамках настоящего исследования будет осуществлена процедура нормализации отношений [23].

В приоритетном порядке проанализируем отношение АПТЕКА в Таблице 5. Примечательным отношением является потому, что в нем нет функциональной, транзитивной зависимости.

Таблица 5 – Отношение АПТЕКА, приведённое к 4НФ

<u>Номер аптеки</u>	<u>Название</u>	<u>Адрес</u>	<u>Номер телефона</u>
1	Твой доктор	Г. Тула, ул. <u>Просп. Ленина</u> , д. 28	88007004403
2	Спар аптека	Г. Тула, ул. Красноармейский проспект д.40	84872343494
3	Столички	Г. Тула, ул. Дмитрия Ульянова д. 2	84872244664

В таблице 5 проведем детальный анализ отношения ИЗГОТОВИТЕЛЬ.

В нем нет функциональной или транзитивной связи.

Таблица 5 – Отношение ИЗГОТОВИТЕЛЬ, приведённое к 4НФ

<u>Номер изготовителя</u>	<u>Название</u>	<u>Адрес</u>	<u>Номер телефона</u>
1	Экохим-инновации	Г. Новомосковск, ул. Дружбы, 8	84876241903
2	АК Синтвита	пос. Шварцевский, ул. Ленина, д. 1	84872564840
3	Тульская фармацевтическая фабрика	Г. Тула, Торховский проезд, д. 10	84872410473

Следующее отношение – ПРЕПАРАТ (Таблица 6). В качестве дополнения появляется атрибут связи, а именно – номер производителя, а также номер аптеки.

Таблица 6 – Отношение ПРЕПАРАТ, приведённое к 1НФ

<u>ID</u> <u>препарата</u>	<u>Противо</u> <u>показания</u> <u>я</u>	<u>Название</u>	<u>Цена</u>	<u>Вес</u>	<u>Название</u> <u>заменителя</u>	<u>Количество</u> <u>о</u>	<u>Срок</u> <u>годности</u>	<u>Отпуск по</u> <u>рецепту</u>	<u>Номер</u> <u>изготовителя</u>	<u>Номер</u> <u>аптеки</u>
1	Сахарный диабет, беременность, детский возраст до 6 лет (для глазных капель).	Називин	170 р	50 гр.	Африн	20	3 года	Необходимо	1	1
2	Сахарный диабет, беременность, детский возраст до 6 лет (для глазных капель).	Нафтизин	200 р	50 гр.	Африн	20	3 года	Необходимо	1	2
3	Сахарный диабет, беременность, детский возраст до 6 лет (для глазных капель).	Виброцил	250 р	50 гр.	Африн	20	3 года	Необходимо	1	3

Следующая задача состоит в том, чтобы адаптировать отношение ПРЕПАРАТ к 4НФ. В таблицах 7 и 8, которые обозначают ПРЕПАРАТ, а также ПРЕПАРАТ-ИЗГОТОВИТЕЛЬ, имеется связь, которая обеспечивается внешним ключом **ID препарата**. Появление ПРЕПАРАТ-ИЗГОТОВИТЕЛЬ в 4НФ не случайно, так как здесь не предусматриваются функциональные, тривиальные и транзитивные зависимости. С учетом того, что отношение ПРЕПАРАТ характеризуется наличием простых атрибутов, его помещают в 1НФ.

Таблица 7 – Отношение ПРЕПАРАТ-ИЗГОТОВИТЕЛЬ, приведённое ко 4НФ

<u>ID препарата</u>	<u>Номер изготовителя</u>
1	1
2	1
3	1

Таблица 8 – Отношение ПРЕПАРАТ, приведённое к 1НФ

<u>ID препарата</u> <u>а</u>	<u>Противопоказания</u>	<u>Название</u>	<u>Цена</u>	<u>Вес</u>	<u>Название заменителя</u>	<u>Количество</u>	<u>Срок годности</u>	<u>Отпуск по рецепту</u>	<u>Номер аптеки</u>
1	Сахарный диабет, беременность, детский возраст до 6 лет (для глазных капель),	Називин	170 р	50 гр.	Африн	20	3 года	Необходимо	1
2	Сахарный диабет, беременность, детский возраст до 6 лет (для глазных капель),	Нафтизин	200 р	50 гр.	Африн	20	3 года	Необходимо	2
3	Сахарный диабет, беременность, детский возраст до 6 лет (для глазных капель),	Виброцил	250 р	50 гр.	Африн	20	3 года	Необходимо	3

Следующая задача состоит в том, чтобы трансформировать отношение ПРЕПАРАТ под 4 НФ. В таблицах 9-10 представлены ПРЕПАРАТ, а также ПРЕПАРАТ-АПТЕКА. В данной ситуации выстраивание связи происходит благодаря тому, что вводится внешний ключ – ID лекарственного средства. Связь ПРЕПАРАТ-АПТЕКА сводится в 4НФ, так как подобных зависимостей, о которых было сказано ранее – нет.

Таблица 9 – Отношение ПРЕПАРАТ, приведённое к 1НФ

<u>ID</u> <u>препарата</u>	<u>Противопоказан</u> <u>ия</u>	<u>Название</u>	<u>Вес</u>	<u>Название</u> <u>заменителя</u>	<u>Срок годности</u>	<u>Отпуск по</u> <u>рецепту</u>
1	Сахарный диабет, беременность, детский возраст до 6 лет (для глазных капель),	Називин	50 гр.	Африн	3 года	Необходимо
2	Сахарный диабет, беременность, детский возраст до 6 лет (для глазных капель),	Нафтизин	50 гр.	Африн	3 года	Необходимо
3	Сахарный диабет, беременность, детский возраст до 6 лет (для глазных капель),	Виброцил	50 гр.	Африн	3 года	Необходимо

Таблица 10 – Отношение ПРЕПАРАТ-АПТЕКА, приведённое к 4НФ

<u>ID препарата</u>	<u>Цена</u>	<u>Количество</u>	<u>Номер аптеки</u>
1	170 р	20	1
2	200 р	20	2
3	250 р	20	3

На рисунке 2 показан процесс нормализации. В результате него были образованы связи, как между первичными, так и внешними ключами. Таким образом, получена вертикальная диаграмма.

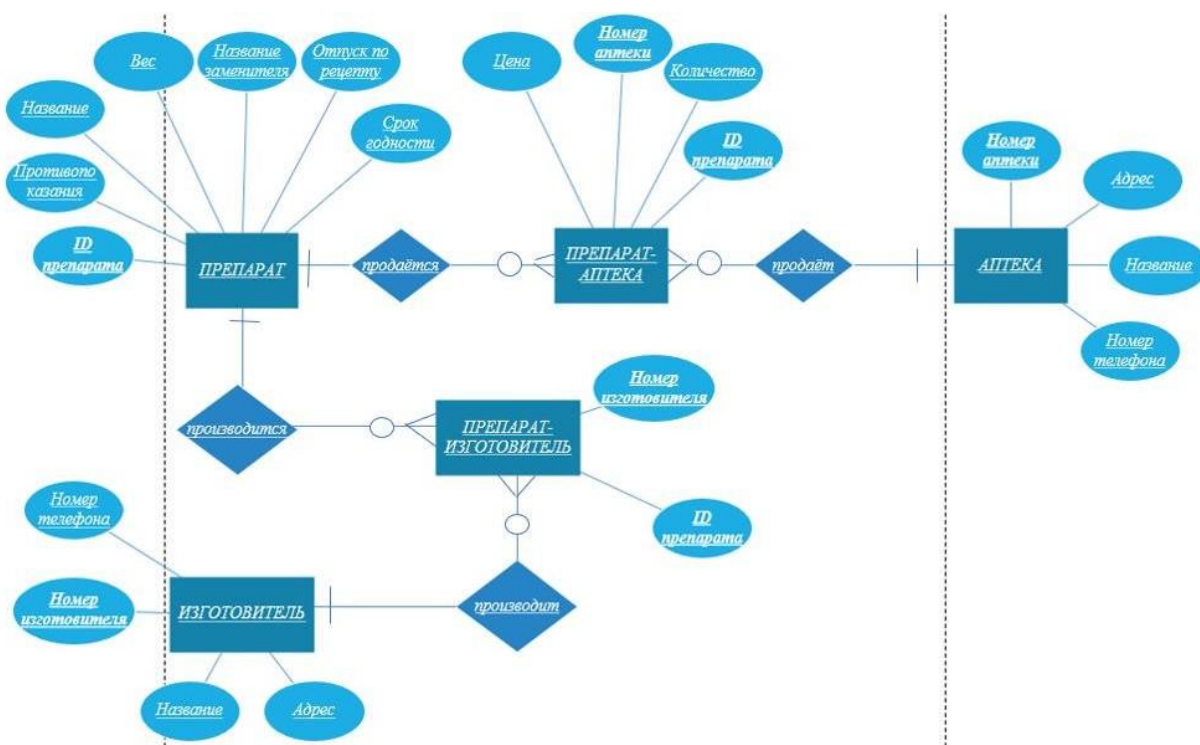


Рисунок 2 - Связи между первичными и внешними ключами в ER-диаграмме после нормализации

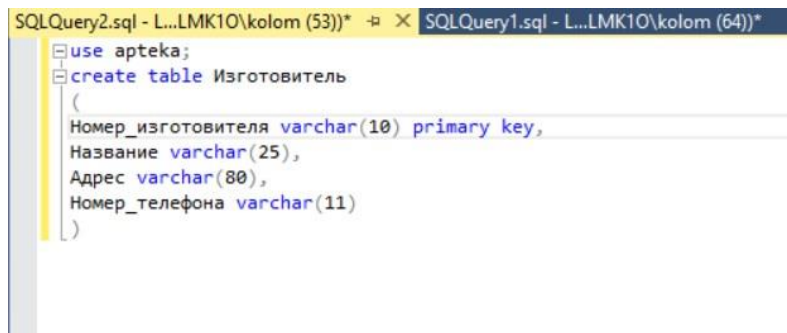
2.5 Физическое проектирование БД

Проектирование в рамках настоящего исследования будет выполняться на БД «Аптека» [15]. Формирование физической модели базируется на логической модели. Далее на рисунке 3 приведен пример, на основании которого осуществляется проектирование.

```
use apteka;
create table Аптека
(
    Номер_аптеки varchar(10) primary key,
    Название varchar(25),
    Адрес varchar(80),
    Номер_телефона varchar(11)
)
```

Рисунок 3 – таблица «Аптека»

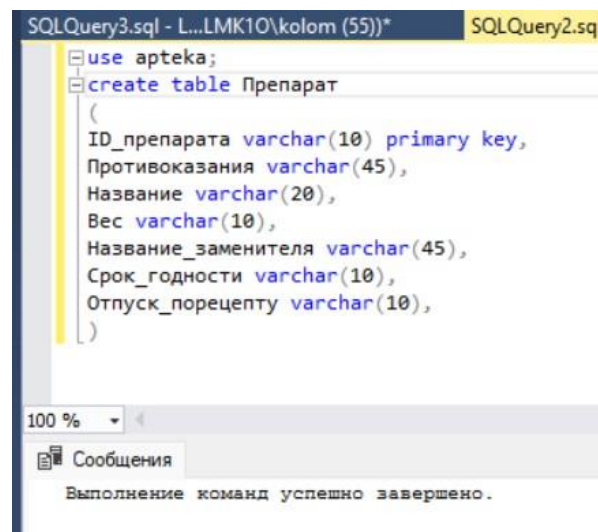
Далее на рисунке 4 проводится проектирование таблицы «Производитель».



```
SQLQuery2.sql - L...LMK10\kolom (53))* SQLQuery1.sql - L...LMK10\kolom (64))*
use apteka;
create table Изготовитель
(
    Номер_изготовителя varchar(10) primary key,
    Название varchar(25),
    Адрес varchar(80),
    Номер_телефона varchar(11)
)
```

Рисунок 4 – таблица «Изготовитель»

Далее на рисунке 5 проводится проектирование таблицы «ЛС».



```
SQLQuery3.sql - L...LMK10\kolom (55))* SQLQuery2.sql
use apteka;
create table Препарат
(
    ID_препарата varchar(10) primary key,
    Противопоказания varchar(45),
    Название varchar(20),
    Вес varchar(10),
    Название_заменителя varchar(45),
    Срок_годности varchar(10),
    Отпуск_порецепту varchar(10),
)
```

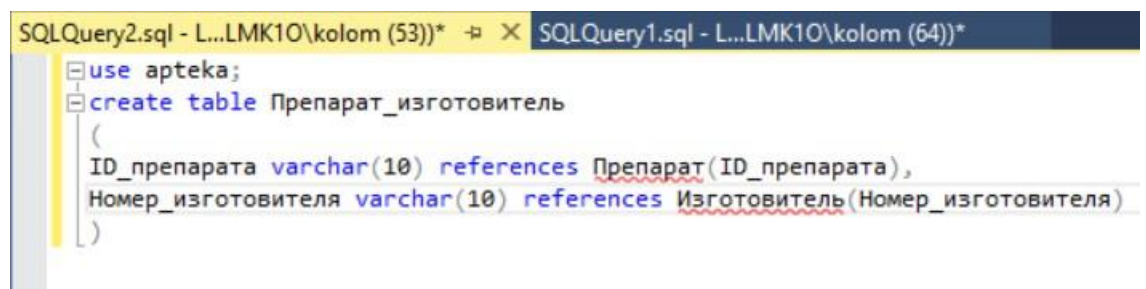
100 %

Сообщения

Выполнение команд успешно завершено.

Рисунок 5 – таблица «ЛС»

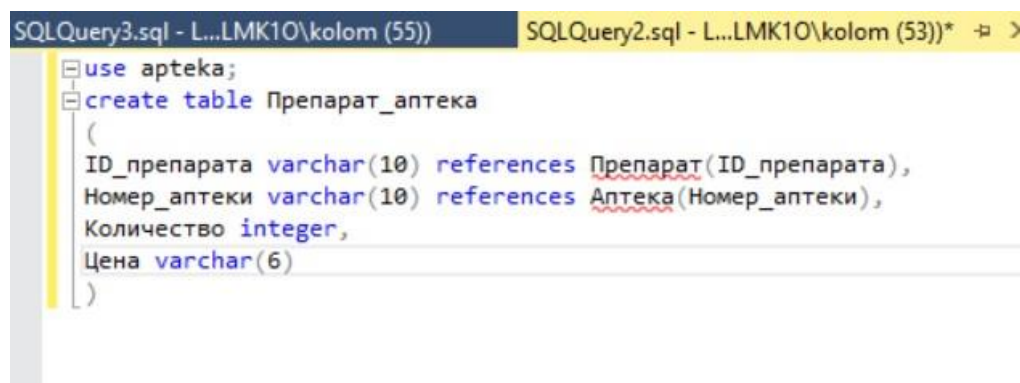
Спроектируем таблицу «Препарат-изготовитель» (Рисунок 6).



```
SQLQuery2.sql - L...LMK10\kolom (53))* SQLQuery1.sql - L...LMK10\kolom (64))*
use apteka;
create table Препарат_изготовитель
(
    ID_препарата varchar(10) references Препарат(ID_препарата),
    Номер_изготовителя varchar(10) references Изготовитель(Номер_изготовителя)
)
```

Рисунок 6 – таблица «Препарат-изготовитель»

Спроектируем таблицу «Препарат-аптека» (Рисунок 7).

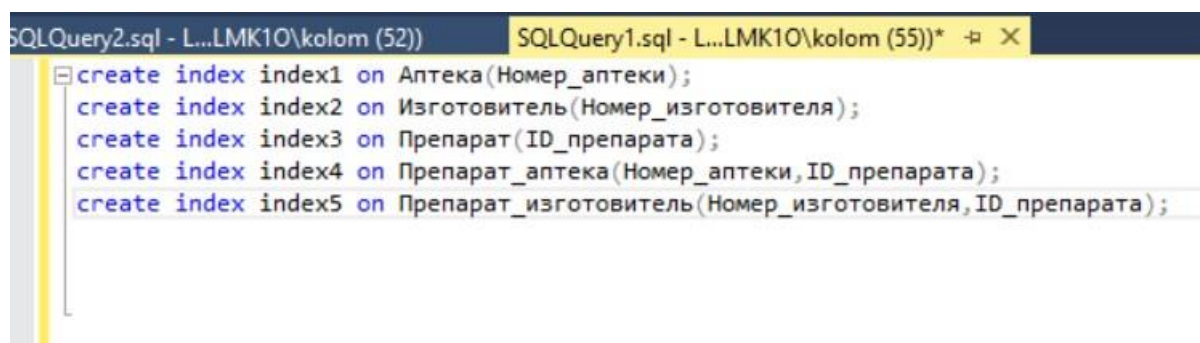


```
SQLQuery3.sql - L...LMK10\kolom (55)) SQLQuery2.sql - L...LMK10\kolom (53))*
use аптека;
create table Препарат_аптека
(
  ID_препарата varchar(10) references Препарат(ID_препарата),
  Номер_аптеки varchar(10) references Аптека(Номер_аптеки),
  Количество integer,
  Цена varchar(6)
)
```

Рисунок 7 – таблица «Препарат-аптека»

Ниже в рамках настоящего исследования проведено формирование индексов под каждую таблицу – рисунок 8. Задача заключается в том, чтобы образовать простые индексы для внесения их в столбцы. Это и будут внешние ключи, которые можно встретить в параметрах поиска. Создание индексов – это задача, которая протекает с применением следующей команды:

«CREATEINDEXtest1_id_indexONtest1 (id);». Очевидно, что применяется индекс «В-дерево».



```
SQLQuery2.sql - L...LMK10\kolom (52)) SQLQuery1.sql - L...LMK10\kolom (55))* X
create index index1 on Аптека(Номер_аптеки);
create index index2 on Изготовитель(Номер_изготовителя);
create index index3 on Препарат(ID_препарата);
create index index4 on Препарат_аптека(Номер_аптеки, ID_препарата);
create index index5 on Препарат_изготовитель(Номер_изготовителя, ID_препарата);
```

Рисунок 8 – Создание индекса на таблицы

Следующая задача состоит в том, чтобы определить, какое потребуется время для прохождения запроса по обновлению, по части добавления, выборки и удаления. Информация расчета приведена в таблицах 11, а также 12.

Таблица 11 – время выполнения запросов без индексации

	INSERT	SELECT	UPDATE	DELETE
Аптека	48 мс	44 мс	45 мс	42 мс
Изготовитель	54 мс	66 мс	52мс	52 мс
Аптека- изготовитель	74 мс	65 мс	69 мс	61 мс
Аптека- препарат	52 мс	41 мс	57 мс	66 мс
Препарат	74 мс	65 мс	69 мс	65 мс

Таблица 12 – Время выполнения запросов с индексацией

	INSERT	SELECT	UPDATE	DELETE
Аптека	34 мс	30 мс	31 мс	31 мс
Изготовитель	42 мс	52 мс	41 мс	45 мс
Аптека- изготовитель	67 мс	45 мс	56 мс	47 мс
Аптека- препарат	45 мс	32 мс	45 мс	54 мс
Препарат	70 мс	60 мс	61 мс	60 мс

По итогам индексации с уверенностью можно утверждать о том, что снижается значительно время в процессе выборки строк. Говоря о выборочности, то по каждому столбцу показатель высокий. Дело в том, что происходит индексация столбцов, первичных ключей, внешних значений, которые неповторимы, уникальные.

2.6 Разработка подсистемы безопасности

Изначально предстоит обозначить локальные сферы [8]. Аптекарь. Этот объект может осуществлять изменение и внесение, просмотр и удаление данных с любой таблицы. Покупателю же таблицы доступны исключительно для просмотра [24].

Что касается роли аптекаря, то она связана с получением прав на изменение полей таблицы в БД, на просмотр информации, как показано на рисунке 9.

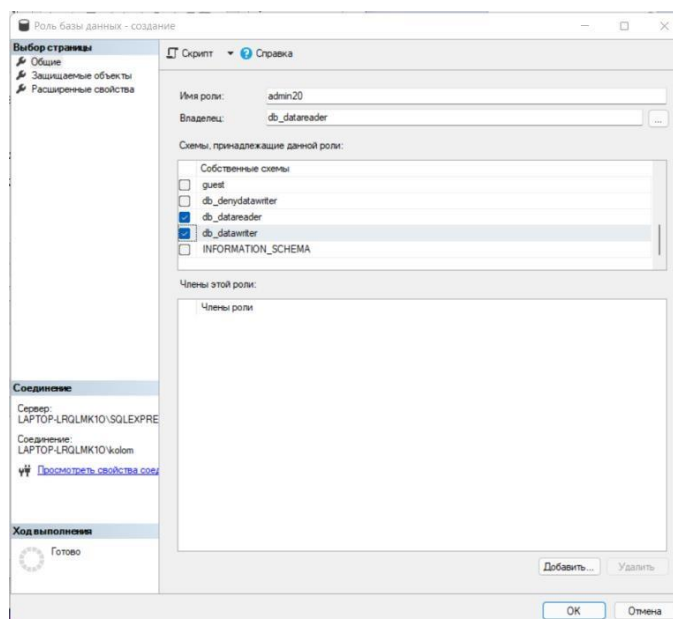


Рисунок 9 – Роль аптекаря

Аптекарь наделяется некоторыми привилегиями. (Рисунок 10).

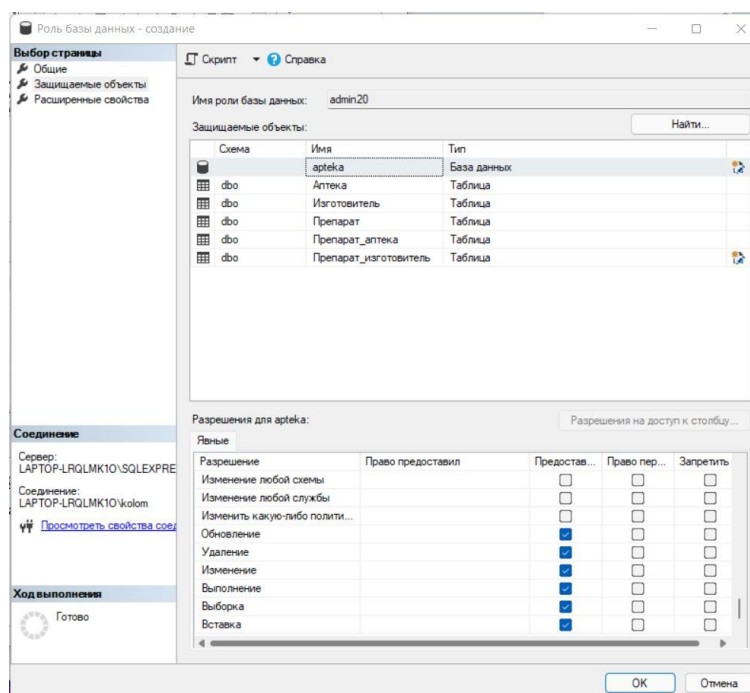


Рисунок 10 – Определение прав аптекаря

Следующая задача состоит в том, чтобы наделить привилегиями покупателя (Рисунок 11).

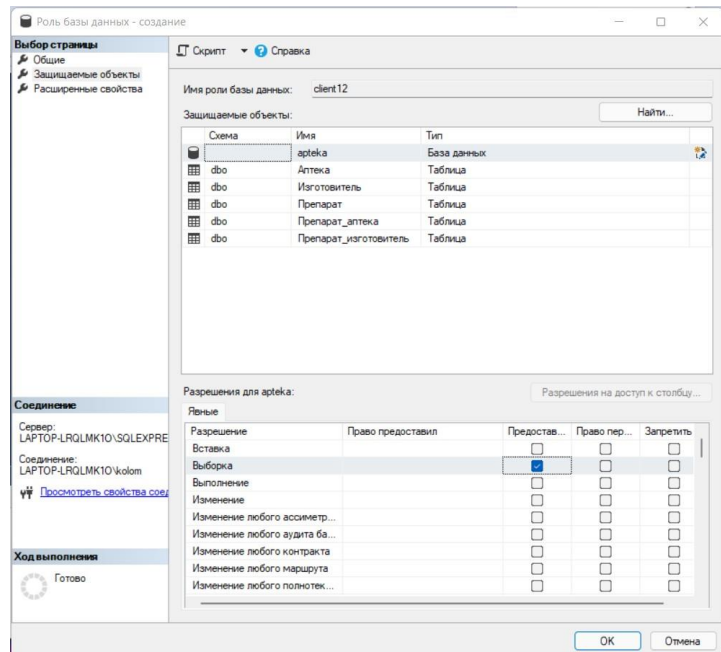


Рисунок 11 – Определение прав покупателя

3 Тестирование реализованного приложения

Проработка проекта осуществляется на основе VisualStudio 2019. Языком программирования для дальнейшей работы становится C#, а вспомогательное средство – шаблон проекта WindowsForms.

Итак, чтобы получить Microsoft.Data.SqlClient, приступаем к его скачиванию во вкладке, которая называется «Управление пакетами» NuGET. Скачиваемый продукт – есть провайдером, который открывает доступ к Майкрософт SQLServer. Вместе с тем, мы получаем отменный драйвер, у которого имеется открытый исходный код. По решению поставщика программы, созданные на языках C#, VisualBasic, F#, имеют доступ к серверу БД. Таким образом, можно подключить таблицы к DataGridView для осуществления разного рода операций [21].

Задается авторизация по форме, чтобы свой доступ имел аптекарь и покупатель – рисунок 12. Вход в систему фармацевта доступен по логину и паролю – идентичен для двух случаев - admin19.

Рисунок 12 – Окно авторизации и вход в окно аптекаря

Аптекарь может работать с любыми таблицами. Первым делом, как показано на рисунке 13 – внесем новый препарат. Далее итог отражается на рисунке 14.

ID_препарата	Противопоказания	Название	Вес	Название_замена	Срок_годности	Отпуск_по_рецепту
1	Сахарный диабе...	Називин	50 гр.	Африн	3 года	Необходимо
2	Сахарный диабе...	Нафтизин	50 гр.	Африн	3 года	Необходимо
3	Сахарный диабе...	Виброцил	50 гр.	Африн	3 года	Необходимо
4	Сахарный диабе...	Виброцил	50 гр.	Африн	4 года	Необходимо

Рисунок 13 – Добавление данных

ID_препарата	Противопоказания	Название	Вес	Название_замена	Срок_годности	Отпуск_по_рецепту
1	Сахарный диабе...	Називин	50 гр.	Африн	3 года	Необходимо
2	Сахарный диабе...	Нафтизин	50 гр.	Африн	3 года	Необходимо
3	Сахарный диабе...	Виброцил	50 гр.	Африн	3 года	Необходимо
4	Сахарный диабе...	Виброцил	50 гр.	Африн	4 года	Необходимо
5	Сахарный диабе...	Нафтизин	50 гр.	Африн	3 года	Необходимо

Рисунок 14 – Новые данные о препаратах

Следующая задача состоит в том, чтобы заменить сведения о препарате; отмечаем в строчке его номер, который нужно удалить или изменить – как показано на рисунке 15. Итог работы на рисунке 16.

ID_препарата	Противопоказания	Название	Вес	Название_заменя	Срок_годности	Отпуск_по_рецепту
1	Сахарный диабе...	Називин	50 гр.	Априн	3 года	Необходимо
2	Сахарный диабе...	Нафтизин	50 гр.	Априн	3 года	Необходимо
3	Сахарный диабе...	Виброцил	50 гр.	Априн	3 года	Необходимо
4	Сахарный диабе...	Виброцил	50 гр.	Априн	4 года	Необходимо
5	Сахарный диабе...	Нафтизин	50 гр.	Априн	3 года	Необходимо

Название препарата(для поиска):

☐ Поиск по номеру
☐ Поиск по названию

Добавить Изменить Удалить Найти Вернуть обратно

Номер препарата: 5
 Противопоказания: Сахарный диабет, бер...
 Название: Нафтизин
 Вес: 50 гр.
 Название_заменя: Априн
 Срок_годности: 3 года
 Отпуск_по_рецепту: Необходимо
 Номер препарата(для поиска/обновления/удаления): 5

Рисунок 15 – Изменение данных строки

ID_препарата	Противопоказания	Название	Вес	Название_заменя	Срок_годности	Отпуск_по_рецепту
1	Сахарный диабе...	Називин	50 гр.	Априн	3 года	Необходимо
2	Сахарный диабе...	Нафтизин	50 гр.	Априн	3 года	Необходимо
3	Сахарный диабе...	Виброцил	50 гр.	Априн	3 года	Необходимо
4	Сахарный диабе...	Виброцил	50 гр.	Априн	4 года	Необходимо
5	Сахарный диабе...	Нафтизин	50 гр.	Априн	4 года	Необходимо

Название препарата(для поиска):

☐ Поиск по номеру
☐ Поиск по названию

Добавить Изменить Удалить Найти Вернуть обратно

Номер препарата:
 Противопоказания:
 Название:
 Вес:
 Название_заменя:
 Срок_годности:
 Отпуск_по_рецепту:
 Номер препарата(для поиска/обновления/удаления):

Рисунок 16 – Результат изменения данных строки

В случае выявления просроченной продукции, проведем ее удаление. Номер 5 будет исходным. Итоги операции отражены на рисунках 17, 18.

ID_препарата	Противопоказания	Название	Вес	Название_замена	Срок_годности	Отпуск_по_рецепту
1	Сахарный диабе...	Називин	50 гр.	Агрин	3 года	Необходимо
2	Сахарный диабе...	Нафтизин	50 гр.	Агрин	3 года	Необходимо
3	Сахарный диабе...	Виброцил	50 гр.	Агрин	3 года	Необходимо
4	Сахарный диабе...	Виброцил	50 гр.	Агрин	4 года	Необходимо
5	Сахарный диабе...	Нафтизин	50 гр.	Агрин	4 года	Необходимо

Рисунок 17 – Удаление данных строки

ID_препарата	Противопоказания	Название	Вес	Название_замена	Срок_годности	Отпуск_по_рецепту
1	Сахарный диабе...	Називин	50 гр.	Агрин	3 года	Необходимо
2	Сахарный диабе...	Нафтизин	50 гр.	Агрин	3 года	Необходимо
3	Сахарный диабе...	Виброцил	50 гр.	Агрин	3 года	Необходимо
4	Сахарный диабе...	Виброцил	50 гр.	Агрин	4 года	Необходимо

Рисунок 18 – Результат удаления данных строки

Строка имеет данные – 1. Препарат называется «Називин». Итог работы на рисунках 19, 20.

Form2

Аптека Изготовитель Препараты Препараты аптека Препараты изготовитель Справка о лекарстве

ID_препарата	Противопоказания	Название	Вес	Название_замен	Срок_годности	Отпуск_порецепт
1	Сахарный диабе...	Називин	50 гр.	Африн	3 года	Необходимо
*						

Название препарата(для поиска)

☒ Поиск по номеру
☐ Поиск по названию

Добавить Изменить Удалить Найти Вернуть обратно

Номер препарата
 Противопоказания
 Название
 Вес
 Название заменителя
 Срок годности
 Отпуск по рецепту
 Номер препарата(для поиска/обновления/удаления)

1

Рисунок 19 – Поиск данных строки по номеру

Form2

Аптека Изготовитель Препараты Препараты аптека Препараты изготовитель Справка о лекарстве

ID_препарата	Противопоказания	Название	Вес	Название_замен	Срок_годности	Отпуск_порецепт
2	Сахарный диабе...	Нафтизин	50 гр.	Африн	3 года	Необходимо
*						

Название препарата(для поиска)

☐ Поиск по номеру
☒ Поиск по названию

Добавить Изменить Удалить Найти Вернуть обратно

Номер препарата
 Противопоказания
 Название
 Вес
 Название заменителя
 Срок годности
 Отпуск по рецепту
 Номер препарата(для поиска/обновления/удаления)

1

Рисунок 20 – Поиск данных строки по названию

На рисунке 21 показан пример, как ведется поиск всех препаратов.

ID_препарата	Противопоказания	Название	Вес	Название_замен.	Срок_годности	Отпуск_по_рецепту
1	Сахарный диабе...	Найтлизин	50 гр.	Априн	3 года	Необходимо
2	Сахарный диабе...	Найтлизин	50 гр.	Априн	3 года	Необходимо
3	Сахарный диабе...	Виброцил	50 гр.	Априн	3 года	Необходимо
4	Сахарный диабе...	Виброцил	50 гр.	Априн	4 года	Необходимо

Найти

Добавить Изменить Удалить Найти Вернуть обратно

Поиск по номеру Поиск по названию

Найтлизин

Номер препарата: _____
 Противопоказания: _____
 Название: _____
 Вес: _____
 Название заменителя: _____
 Срок годности: _____
 Отпуск по рецепту: _____
 Номер препарата (для поиска/обновления/удаления): 1

Рисунок 21 – Поиск всех препаратов

Позицию 5 нужно перенести в аптеку 3. Цена товара будет 250 рублей. Кроме прочего, в указанную аптеку будет также распределено 50 других позиций лекарств – рисунки 22, 23.

ID_препарата	Номер_аптеки	Количество	Цена
1	1	20	170
2	2	20	200
3	4	20	250
4	4	25	250

Найти

Добавить Изменить Удалить Найти

Номер препарата: 5
 Номер аптеки: 3
 Количество: 50
 Цена: 250

Номер препарата (для поиска/обновления/удаления): _____
 Номер аптеки (для поиска/обновления/удаления): _____

Вернуть обратно

Поиск по номеру препарата Поиск по номеру аптеки

Рисунок 22 – Добавление данных

The screenshot shows the 'Form2' application window. It has a menu bar with 'Аптека', 'Изготовитель', 'Препараты', 'Препараты аптека', 'Препараты изготовитель', and 'Справка о лекарстве'. The main area contains a table with the following data:

ID_препарата	Номер_аптеки	Количество	Цена
1	1	20	170
2	2	20	200
3	4	20	250
4	4	25	250
5	3	50	250

Below the table are four buttons: 'Добавить', 'Изменить', 'Удалить', and 'Найти'. To the right of the table is a form with the following fields:

- Номер препарата: [input field]
- Номер аптеки: [input field]
- Количество: [input field]
- Цена: [input field]
- Номер препарата (для поиска/обновления/удаления): [input field]
- Номер аптеки(для поиска/обновления/удаления): [input field]

Below the form is a 'Вернуть обратно' button and two radio buttons: 'Поиск по номеру препарата' and 'Поиск по номеру аптеки'.

Рисунок 23 – Новые данные

Препарат под номером 5 меняется в цене, как показано на рисунке 24, и рисунке 25.

The screenshot shows the 'Form2' application window. The table data is the same as in Figure 23, but the price for the medicine with ID 5 has been updated to 350:

ID_препарата	Номер_аптеки	Количество	Цена
1	1	20	170
2	2	20	200
3	4	20	250
4	4	25	250
5	3	50	350

The form on the right now contains the following values:

- Номер препарата: 5
- Номер аптеки: 3
- Количество: 50
- Цена: 350

The 'Вернуть обратно' button and radio buttons remain the same.

Рисунок 24 – Изменение данных строки

ID_препарата	Номер_аптеки	Количество	Цена
1	1	20	170
2	2	20	200
3	4	20	250
4	4	25	250
5	3	50	350

Рисунок 25 – Результат изменения данных строки

Интерфейс, рабочие процессы во вкладках АПТЕКА, ЛС, ПРОИЗВОДИТЕЛЬ – точно такие, как и на других вкладках. Примеры будут приведены на рисунках 26-28.

Номер_аптеки	Название	Адрес	Номер_телефона
1	Твой доктор	Г. Тула, ул. Про...	88007004403
2	Спар аптека	Г. Тула, ул. Кра...	84872343494
3	Столики	Г. Тула, ул. Дни...	84872244664
4	Столики	Г. Тула, ул. Дни...	84872244664

Рисунок 26 – Вкладка «Аптека»

Номер_изготовит	Название	Адрес	Номер_телефона
1	Экохиминноваци...	г. Новомосковск...	84876241903
2	АК Синтема	пос. Шарьевск...	84872564840
3	Тульская фарм...	г. Тула, Торков...	84872410473

Рисунок 27 – Вкладка «Изготовитель»

ID_препарата	Номер_изготовит
1	1
2	1
3	3
4	3
5	3

Рисунок 28 – Вкладка «Препараты изготовитель»

На рисунках 29 и 30 показана процедура оформления справки пациенту.

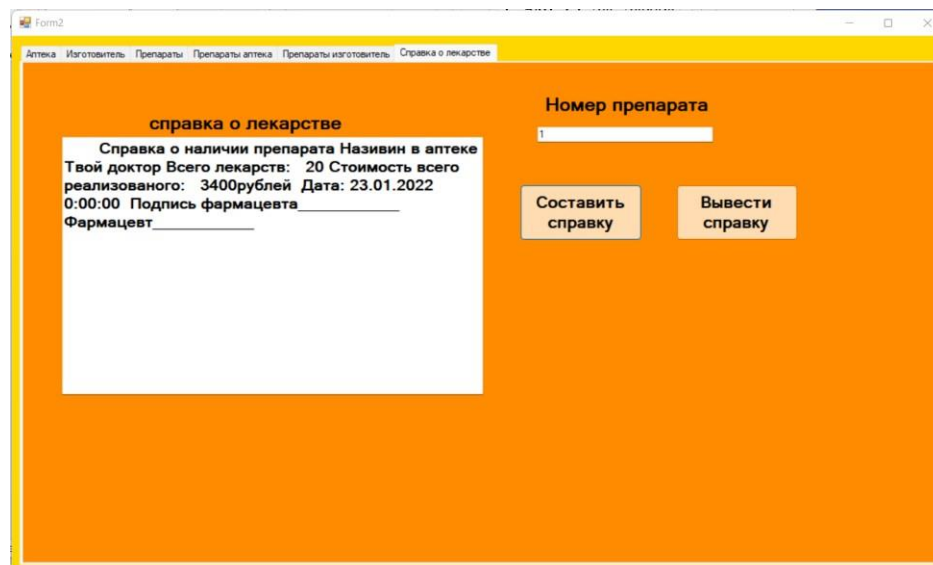
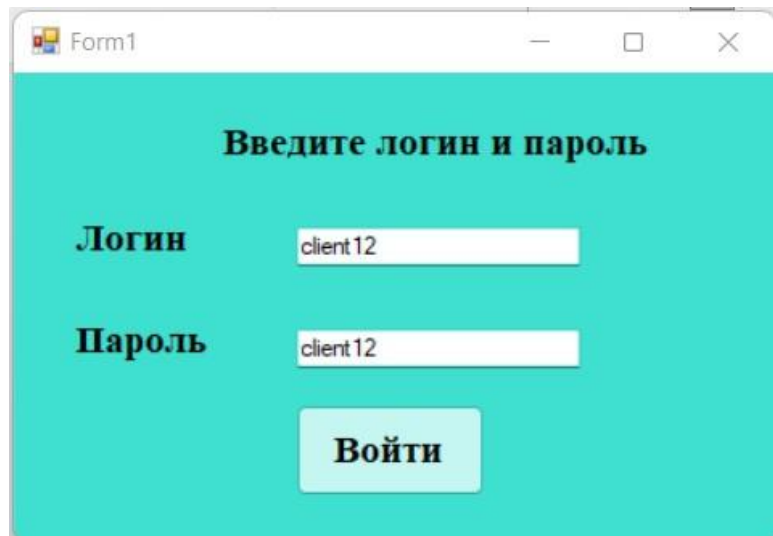


Рисунок 29 – создание справки

Справка о наличии препарата Називин
в аптеке Твой доктор
Всего лекарств: 20
Стоимость всего реализованного: 3400рублей
Дата: 23.01.2022 0:00:00
Подпись фармацевта _____
Фармацевт _____

Рисунок 30 – вывод справки

В окне, предназначенном для клиента, вводится логин и пароль «client12», как показано на рисунке 31.



Введите логин и пароль

Логин

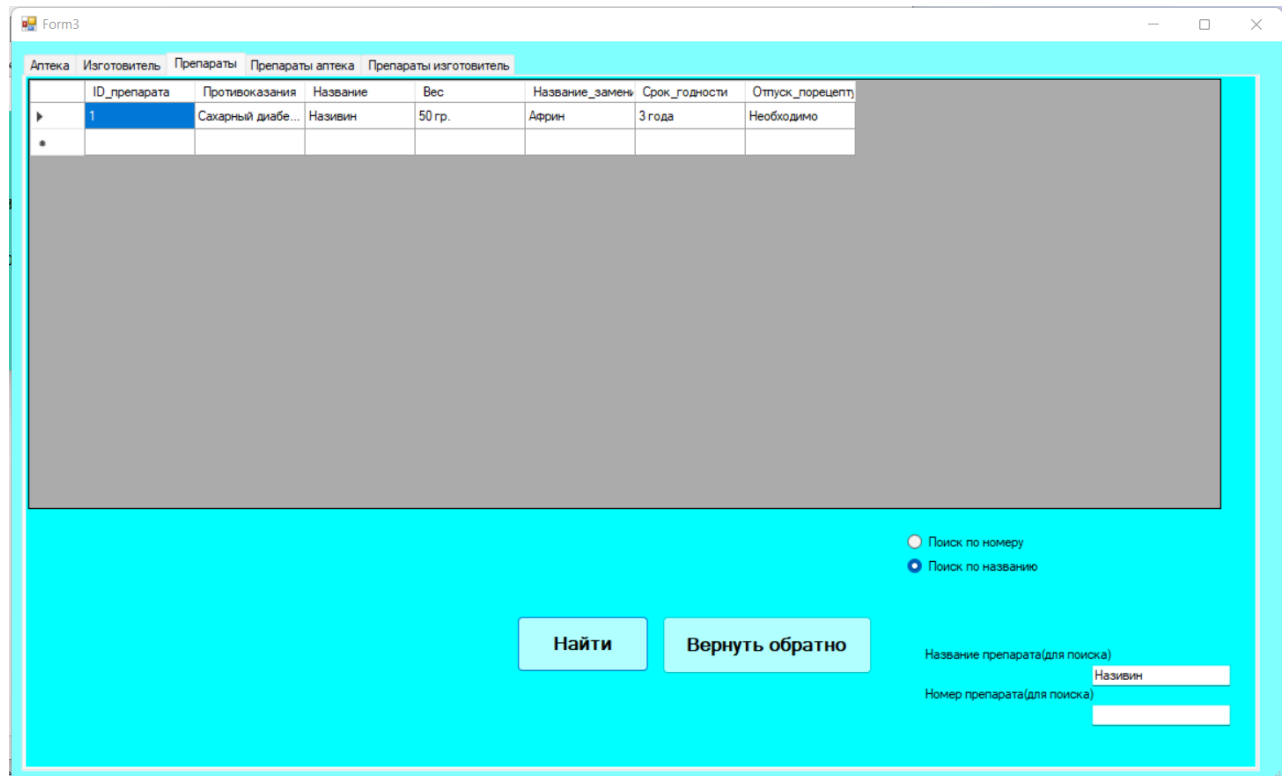
Пароль

Войти

Рисунок 31 – Вход в окно клиента

Напомним, что покупатель просто может смотреть таблицы.

Отметим препарат «Називин», как показано на рисунке 32.



ID_препарата	Противопоказания	Название	Вес	Название_замен	Срок_годности	Отпуск_порецпт
1	Сахарный диабе...	Називин	50 гр.	Адрин	3 года	Необходимо

☒ Поиск по номеру
☐ Поиск по названию

Название препарата(для поиска)
 Номер препарата(для поиска)

Рисунок 32 – выборка препарата по названию

На рисунке 33 показано, как все вернуть в исходное состояние.

The screenshot shows a window titled 'Form3' with a menu bar containing 'Аптека', 'Изготовитель', 'Препараты', 'Препараты аптеки', and 'Препараты изготовитель'. Below the menu is a table with the following data:

ID_препарата	Противопоказания	Название	Вес	Название_замен	Срок_годности	Отпуск_порецшт
1	Сахарный диабе...	Наплизин	50 гр.	Акрил	3 года	Необходимо
2	Сахарный диабе...	Наплизин	50 гр.	Акрил	3 года	Необходимо
3	Сахарный диабе...	Виброцил	50 гр.	Акрил	3 года	Необходимо
4	Сахарный диабе...	Виброцил	50 гр.	Акрил	4 года	Необходимо
5	Сахарный диабе...	Наплизин	50 гр.	Акрил	3 года	Необходимо

Below the table is a large grey rectangular area. At the bottom right, there are two radio buttons: 'Поиск по номеру' (selected) and 'Поиск по названию'. Below these are two buttons: 'Найти' and 'Вернуть обратно'. At the bottom right, there are two input fields: 'Название препарата(для поиска)' and 'Номер препарата(для поиска)'.

Рисунок 33– Выборка обратно

Следующая задача состоит в том, чтобы найти аптеку, в которой продается требуемое ЛС. Также можно найти прочие сведения о торговой точке, о лекарстве – рисунки 34, 35.

The screenshot shows a window titled 'Form3' with a menu bar containing 'Аптека', 'Изготовитель', 'Препараты', 'Препараты аптеки', and 'Препараты изготовитель'. Below the menu is a table with the following data:

ID_препарата	Номер_аптеки	Количество	Цена
1	1	20	170

Below the table is a large grey rectangular area. On the right side, there are two input fields: 'Номер препарата (для поиска)' and 'Номер аптеки(для поиска)'. Below these is a button 'Вернуть обратно'. At the bottom right, there are two radio buttons: 'Поиск по номеру препарата' (selected) and 'Поиск по номеру аптеки'. Below these is a button 'Найти'.

Рисунок 34 – просмотр информации об аптеке где реализовывается препарат

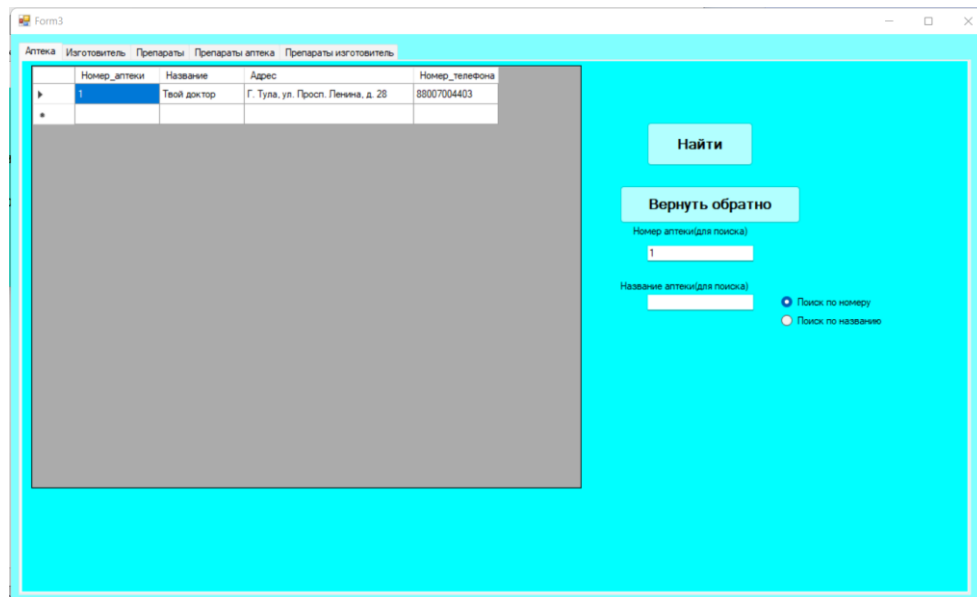


Рисунок 35 – просмотр информации об аптеке

Нужный препарат ищется по алгоритму, который отражен на рисунках 36 и 37

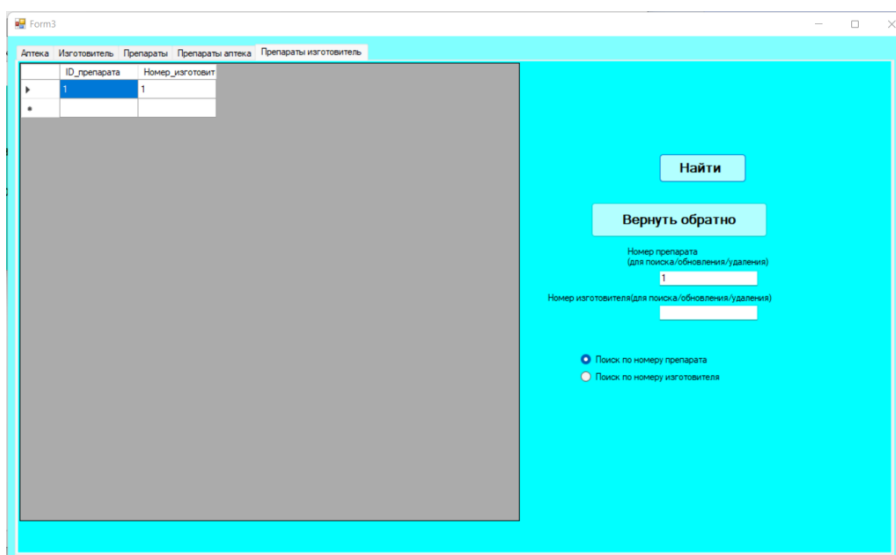


Рисунок 36 – просмотр информации об изготовителе данного препарата

Form3

Аптека Изготовитель Препараты Препараты аптеки Препараты изготовитель

Номер_изготовит.	Название	Адрес	Номер_телефона
1	Экспресс-диагност.	Г. Новооскольск...	84876241903

Найти

Вернуть обратно

Номер изготовителя(для поиска)
1

Название изготовителя(для поиска)

☒ Поиск по номеру
☐ Поиск по названию

Рисунок 37 – просмотр информации об изготовителе

4 Оценка и обоснование экономической эффективности разработки информационной системы

Теперь стоит задача, которая предусматривает оценку прямого экономического эффекта, который можно получить после проработки, а также интеграции нашей разработки на складе аптеки. Трудовые затраты за месяц будут следующими.

Так как рабочих дней в неделе – 5, в месяце – 4 рабочие недели, то с учетом потребности в 40 отчетов проводится оценка трудовых затрат программиста на создание соответствующих документов. Будем считать, что создание, передача, сохранение отчета – это процедуры, на реализацию которых требуется 5 минут. Следовательно, базовый вариант разработки за 1 месяц потребует 4000 минут. С учетом того, что работа администратора по времени точно такая, как и у программиста, проведем расчет трудовых затрат, которые будут направлены на автоматизацию решения. Если трудовые затраты по администрированию БД – 1 час в день, то для всего проекта за месяц тратится 1200 минут времени в работе администратора.

Теперь переходим к непосредственному расчету трудовых затрат. Формула, по которой определяется показатель ΔT приведена далее:

$$\Delta T = T_0 - T_1, \quad (1)$$

где T_0 – временные расходы по части основного бизнес-процесса;

T_1 – требуемое время для проектного варианта.

Понижение трудовых затрат составляет 2800 минут.

Коэффициент понижения трудовых затрат K_T
(в процентах), рассчитаем по формуле:

$$K_T = \frac{\Delta T}{T_0} \times 100\%. \quad (2)$$

Для проектного решения аналогичный коэффициент составляет 70%.

Индекс понижения трудовых затрат Y_T помогает рассчитать уравнение:

$$Y_T = \frac{T_0}{T}. \quad (3)$$

Получаем значение, равное 3,34.

Очередная задача заключается в том, чтобы определить стоимостные параметры затрат, которые будут сопряжены с реализацией бизнес-процесса. Итак, стоимостные параметры рассчитываются допущением, согласно которому цена на единицу времени администратора аналогична такому же показателю у программиста, а именно – 200,000 рублей за 160 часов работы. Соответственно, 1 минута работы обойдется в 20,48 рублей.

Потребуется также покупка техники. В затраты по проектному варианту включается амортизация. Каждый месяц по ней приходится сумма, определяемая по формуле:

$$\Sigma_M = \frac{C_{\text{п}}}{n}, \quad (4)$$

где $C_{\text{п}}$ – рыночная цена;

n – полезное действие, в мес.

Для срока полезного действия, который составляет 3 года, и с начальной стоимости техники в 55000 рублей, имеем амортизацию каждый месяц в сумме 1527,78 рублей. Затраты на 1 месяц работы в базовом варианте C_0 такие же, как в случае с ценой на 1 минуту труда, перемноженное на T_0 затрат на базовый вариант. То есть, получаем сумму в 81920 рублей. Для проектного варианта сумма расходов достигает 26103,78 рублей за 1 месяц. Теперь непосредственно определяется стоимость абсолютных затрат:

$$\Delta C = C_0 - C_1, \quad (5)$$

где C_0 – расходы по базовому варианту

C_1 – проектные расходы.

Таким образом, получаем 55816,22 рублей за каждый месяц.

Коэффициент относительного понижения стоимостных расходов K_C (в процентах) найдем по формуле:

$$K_C = \frac{\Delta C}{C_0} \times 100\%. \quad (6)$$

Следовательно, можно утверждать о том, что для нашего проекта коэффициент понижения стоимостных затрат составляет 0,68.

Индекс снижения стоимостных затрат Y_C выражен равенством:

$$Y_C = \frac{C_0}{C_1}. \quad (7)$$

В нашем случае имеем величину 3,14.

Ниже в таблице 13 будут приведены параметры, согласно которым ведется расчет экономической эффективности интеграции проекта ИС.

Таблица 13 – Показатели экономической эффективности внедрения проекта информационной системы

Трудоёмкость	Затраты		ΔT (1)	K_T (2)	Y_T (3)
	Базовый вариант	Проектный вариант			
	T_0 (час)	T_1 (час)			
	66,67	20	46,67	70%	3,34
Стоимость	C_0 (руб.)	C_1 (руб.)	ΔC (5)	K_C (6)	Y_C (7)
	81920	26103,78	55816,22	0,68	3,14

Получается, что Y_{Ti} и Y_{Ci} имеют небольшое расхождение, что обуславливается введением такого дополнительного параметра, как амортизационные отчисления.

На проект капитальные расходы также предусматривают стоимость техники, стоимость ее монтажа, настройки. Реализовать проект сможет даже 1 специалист, в течение 2 месяцев. Таким образом, ему нужно заплатить 400 тысяч рублей. Кроме прочего, в состав капитальных затрат нужно ввести стоимость часового разового обучения пользователей. Принимая во внимание тот факт, что менеджер по продажам имеет зарплату в 150 тысяч рублей в месяц, час обучения его обойдется компании в 937,5 рублей. Оптимальная численность лиц в группе, с которой будет работать администратор – 20 человек. Всего же системой пользуется 40 человек. Эти данные позволяют определить цену интеграции проектного решения. Имеем 39957,6 рублей.

Капитальные затраты по проекту, в целом составляют 494957,60 рублей.

Теперь определим окупаемость проекта T_{OK}

$$T_{OK} = \frac{K_{\Pi}}{\Delta C} \quad (8)$$

Заключение

В завершение работы можно выделить следующее:

- В главе 1 был изучен обзор путей и средств достижения цели. А также по работе были рассмотрены такие средства как: MSSQLServer, Visio и VisualStudio. Благодаря ним удастся достичь поставленных целей

- В главе 2 было произведено проектирование АИС по учёту лекарственных средств. Было произведено инфологическое проектирование, проведена оценка операционной обстановки, произведено логическое проектирование БД и по нему было произведено физическое проектирование БД. Из всего этого можно сделать вывод о том, что проектирование АИС прошло успешно.

В главе 3 было произведено тестирование АИС на функционал. Следовательно можно сделать вывод о том, что приложение работает достаточно хорошо.

Наконец, четвертая глава настоящего исследования – это оценка экономического эффекта, которого удастся достичь предприятию после проработки, а также комплексной интеграции проекта в функционировании склада. Проводится расчет, как стоимостных, так и трудовых затрат; вместе с тем, вынесена оценка совершаемым затратам, которые были направлены на потребности проектирования. Особое внимание уделяется срокам окупаемости проекта.

Закономерный результат исследования – проработка приложения, с помощью которого можно выполнять работу с БД «АИС по учету ЛС на складе аптеки».

Список используемых источников

1. Автоматизация проектирования вычислительных систем. Языки, моделирование и базы данных / ред. М. Брейер. - М.: Мир, 2020. - 19 с.
2. Алексей, Горшков undОлжас Исаков Вопросы системного проектирования крупных информационных систем / Алексей Горшков undОлжас Исаков. - М.: LAP LambertAcademicPublishing, 2019. - 35 с.
3. Благодаров, Андрей Витальевич Алгоритмы категорирования персональных данных для систем автоматизированного проектирования баз данных информационных систем: моногр. / Благодаров Андрей Витальевич. - М.: Горячая линия - Телеком, 2019. - 12 с.
4. Боковой, Ю. В. Особенности методологии проектирования информационных систем для малого и среднего бизнеса / Ю.В. Боковой. - М.: Синергия, 2021. - 17 с.
5. Вендров, А.М. Практикум по проектированию программного обеспечения экономических информационных систем. Гриф МО РФ / А.М. Вендров. - М.: Финансы и статистика, 2020. - 28 с.
6. Вичугова, А. А. Методы и средства концептуального проектирования информационных систем: сравнительный анализ структурного и объектно-ориентированного подходов / А.А. Вичугова. - М.: Синергия, 2018. - 5 с.
7. Воройский, Ф. С. Основы проектирования автоматизированных библиотечно-информационных систем / Ф.С. Воройский. - М.: ФИЗМАТЛИТ, 2021. - 8 с.
8. Зегжда, Д.П. Основы безопасности информационных систем / Д.П. Зегжда, А.М. Ивашко. - М.: Горячая линия - Телеком, 2018. - 21 с.
9. Ипатова, Э. Р. Методологии и технологии системного проектирования информационных систем / Э.Р. Ипатова, Ю.В. Ипатов. - М.: Флинта, 2020. - 29 с.
10. Кутикова, К. В. Методика проектирования информационных систем для сферы государственных и муниципальных услуг / К.В. Кутикова. - М.: Синергия, 2018. - 23 с.

11. Мартишин, С.А. Базы данных. Практическое применение СУБД SQL и NoSQL-типа для применения проектирования информационных систем. Гриф МО РФ: моногр. / С.А. Мартишин. - М.: Инфра-М, Форум, 2018. - 38 с.
12. Мартишин, С.А. Базы данных. Практическое применение СУБД SQL и NoSQL-типа для применения проектирования информационных систем. Учебное пособие. Гриф МО РФ / С.А. Мартишин. - М.: Инфра-М, Форум, 2021. - 34 с.
13. Мацяшек, Лешек А. Анализ и проектирование информационных систем с помощью UML 2.0 / Мацяшек Лешек А.. - М.: Диалектика / Вильямс, 2018. - 8 с.
14. Мидоу, Ч. Анализ информационных систем / Ч. Мидоу. - М.: Прогресс, 2020. - 19 с.
15. Подчукаев, Владимир Автоматическое проектирование информационно-управляющих систем / Владимир Подчукаев. - М.: LAP LambertAcademicPublishing, 2019. - 19 с.
16. Путилин, А. Б. Компонентное моделирование и программирование на языке UML. Практическое руководство по проектированию информационных систем / А.Б. Путилин, Е.А. Юрагов. - М.: НТ Пресс, 2021. - 30 с.
20. Раскин, Д. Интерфейс: Новые направления в проектировании компьютерных систем / Д. Раскин. - М.: Символ-плюс, 2019. - 26 с.
21. Слепцов, А.И. Автоматизация проектирования управляющих систем гибких автоматизированных производств: моногр. / А.И. Слепцов, А.А. Юрасов. - М.: Техніка, 2021. - 39 с.
22. Филиппов, В. А. Многомерные СУБД при создании корпоративных информационных систем / В.А. Филиппов. - М.: Едиториал УРСС, 2020. - 3 с.
23. Шоу, А. Логическое проектирование операционных систем / А. Шоу. - М.: Мир, 2021. - 35 с.
24. Ярочкин, В. Безопасность информационных систем / В. Ярочкин. - М.: Ось-89, 2018. - 14 с.

Приложение А

Листинг 1 – Form1.cs

Листинг 1 – Form1.cs

```
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 БД_аптека
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            private void button1_Click(object sender, EventArgs e)
            {
                string AdmPass = "admin19";
                string AdmLgn = "admin19";
                string КлиентPass = "client12";
                string КлиентLgn = "client12";

                if (textBox1.Text == AdmLgn && textBox2.Text == AdmPass)
                {
                    Form2 admin = new Form2();
                    admin.ShowDialog();
                    textBox1.Clear();
                    textBox2.Clear();
                }

                if (textBox1.Text == КлиентLgn && textBox2.Text == КлиентPass)
                {
                    Form3 reader = new Form3();
                    reader.ShowDialog();
                    textBox1.Clear();
                    textBox2.Clear();
                }
            }
        }
    }
}
```

Приложение Б

Листинг 2 – Form2.cs

Листинг 2 – Form2.cs

```
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 БД_антека
{
    public partial class Form2 : Form
    {
        Dtdb bd = new Dtdb();
        public Form2()
        {
            InitializeComponent();
            bd.ShowАптека(dataGridView1);
            bd.ShowИзготовитель(dataGridView2);
            bd.ShowПрепараты(dataGridView3);
            bd.ShowПрепараты_антека(dataGridView4);
            bd.ShowПрепараты_изготовитель(dataGridView5);
        }

        private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
        {
            DataGridViewRow fill = dataGridView1.CurrentRow;
            textBox1.Text = Convert.ToString(fill.Cells[0].Value);
            textBox2.Text = Convert.ToString(fill.Cells[1].Value);
            textBox3.Text = Convert.ToString(fill.Cells[2].Value);
            textBox21.Text = Convert.ToString(fill.Cells[3].Value);
        }

        private void dataGridView2_CellContentClick(object sender, DataGridViewCellEventArgs e)
        {
            DataGridViewRow fill = dataGridView2.CurrentRow;
            textBox4.Text = Convert.ToString(fill.Cells[0].Value);
            textBox5.Text = Convert.ToString(fill.Cells[1].Value);
            textBox6.Text = Convert.ToString(fill.Cells[2].Value);
            textBox7.Text = Convert.ToString(fill.Cells[3].Value);
        }

        private void dataGridView3_CellContentClick(object sender, DataGridViewCellEventArgs e)
        {
            DataGridViewRow fill = dataGridView3.CurrentRow;
            textBox8.Text = Convert.ToString(fill.Cells[0].Value);
            textBox9.Text = Convert.ToString(fill.Cells[1].Value);
            textBox10.Text = Convert.ToString(fill.Cells[2].Value);
            textBox11.Text = Convert.ToString(fill.Cells[3].Value);
            textBox12.Text = Convert.ToString(fill.Cells[4].Value);
            textBox13.Text = Convert.ToString(fill.Cells[5].Value);
            textBox14.Text = Convert.ToString(fill.Cells[6].Value);
        }
    }
}
```

Продолжение Приложения Б

Продолжение Листинг 2 – Form2.cs

```
private void dataGridView4_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    DataGridViewRow fill = dataGridView4.CurrentRow;
    textBox15.Text = Convert.ToString(fill.Cells[0].Value);
    textBox16.Text = Convert.ToString(fill.Cells[1].Value);
    textBox17.Text = Convert.ToString(fill.Cells[2].Value);
    textBox18.Text = Convert.ToString(fill.Cells[3].Value);
}

private void dataGridView5_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    DataGridViewRow fill = dataGridView5.CurrentRow;
    textBox19.Text = Convert.ToString(fill.Cells[0].Value);
    textBox20.Text = Convert.ToString(fill.Cells[1].Value);
}

private void button5_Click(object sender, EventArgs e)
{
    bd.ShowАптека(dataGridView1);
}

private void button6_Click(object sender, EventArgs e)
{
    bd.ShowИзготовитель(dataGridView2);
}

private void button11_Click(object sender, EventArgs e)
{
    bd.ShowПрепараты(dataGridView3);
}

private void button30_Click(object sender, EventArgs e)
{
    bd.ShowПрепараты_аптека(dataGridView4);
}

private void button20_Click(object sender, EventArgs e)
{
    bd.ShowПрепараты_изготовитель(dataGridView5);
}

private void button4_Click(object sender, EventArgs e)
{
    if (radioButton1.Checked == true && radioButton2.Checked == false)
    {
        bd.ShowПоиск_Аптека(dataGridView1, textBox25.Text);
    }
    if (radioButton2.Checked == true && radioButton1.Checked == false)
    {
        bd.ShowПоиск_Аптека1(dataGridView1, textBox22.Text);
    }
}

private void textBox4_TextChanged(object sender, EventArgs e)
{
}

private void button7_Click(object sender, EventArgs e)
{
    if (radioButton4.Checked == true && radioButton3.Checked == false)
    {
        bd.ShowПоиск_Изготовитель(dataGridView2, textBox26.Text);
    }
    if (radioButton3.Checked == true && radioButton4.Checked == false)
    {
        bd.ShowПоиск_Изготовитель1(dataGridView2, textBox23.Text);
    }
}
```

Продолжение Листинг 2 – Form2.cs

```

    }

    private void button12_Click(object sender, EventArgs e)
    {
        if (radioButton6.Checked == true && radioButton5.Checked == false)
        {
            bd.ShowПоиск_Препарат(dataGridView3, textBox27.Text);
        }
        if (radioButton5.Checked == true && radioButton6.Checked == false)
        {
            bd.ShowПоиск_Препарат1(dataGridView3, textBox24.Text);
        }
    }

    private void button16_Click(object sender, EventArgs e)
    {
        if (radioButton8.Checked == true && radioButton7.Checked == false)
        {
            bd.ShowПоиск_Препарат_аптека(dataGridView4, textBox28.Text);
        }
        if (radioButton7.Checked == true && radioButton8.Checked == false)
        {
            bd.ShowПоиск_Препарат_аптека1(dataGridView4, textBox31.Text);
        }
    }

    private void button21_Click(object sender, EventArgs e)
    {
        if (radioButton10.Checked == true && radioButton9.Checked == false)
        {
            bd.ShowПоиск_Препарат_изготовитель(dataGridView5, textBox29.Text);
        }
        if (radioButton9.Checked == true && radioButton10.Checked == false)
        {
            bd.ShowПоиск_Препарат_изготовитель1(dataGridView5, textBox32.Text);
        }
    }

    private void button1_Click(object sender, EventArgs e)
    {
        string n1 = textBox1.Text;
        string n2 = textBox2.Text;
        string n3 = textBox3.Text;
        string n4 = textBox21.Text;
        bd.AddAnteka(n1, n2, n3, n4);
        textBox1.Text = "";
        textBox2.Text = "";
        textBox3.Text = "";
        textBox21.Text = "";
        bd.ShowAnteka(dataGridView1);
    }

    private void button2_Click(object sender, EventArgs e)
    {
        string n1 = textBox1.Text;
        string n2 = textBox2.Text;
        string n3 = textBox3.Text;
        string n4 = textBox21.Text;
        string n5 = textBox25.Text;
        bd.UpdAnteka(n1, n2, n3, n4, n5);
        textBox1.Text = "";
        textBox2.Text = "";
        textBox3.Text = "";
        textBox21.Text = "";
        textBox25.Text = "";
        bd.ShowAnteka(dataGridView1);
    }
}

```

Продолжение Листинг 2 – Form2.cs

```
private void button3_Click(object sender, EventArgs e)
{
    string n5 = textBox25.Text;
    bd.DelAnteka(n5);
    textBox25.Text = "";
    bd.ShowAnteka(dataGridView1);
}

private void button10_Click(object sender, EventArgs e)
{
    string n1 = textBox4.Text;
    string n2 = textBox5.Text;
    string n3 = textBox6.Text;
    string n4 = textBox7.Text;
    bd.AddИзготовитель(n1, n2, n3, n4);
    textBox4.Text = "";
    textBox5.Text = "";
    textBox6.Text = "";
    textBox7.Text = "";
    bd.ShowИзготовитель(dataGridView2);
}

private void button9_Click(object sender, EventArgs e)
{
    string n1 = textBox4.Text;
    string n2 = textBox5.Text;
    string n3 = textBox6.Text;
    string n4 = textBox7.Text;
    string n5 = textBox26.Text;
    bd.UpdИзготовитель(n1, n2, n3, n4, n5);
    textBox4.Text = "";
    textBox5.Text = "";
    textBox6.Text = "";
    textBox7.Text = "";
    textBox26.Text = "";
    bd.ShowИзготовитель(dataGridView2);
}

private void button8_Click(object sender, EventArgs e)
{
    string n5 = textBox26.Text;
    bd.DelИзготовитель(n5);
    textBox26.Text = "";
    bd.ShowИзготовитель(dataGridView2);
}

private void button15_Click(object sender, EventArgs e)
{
    string n1 = textBox8.Text;
    string n2 = textBox9.Text;
    string n3 = textBox10.Text;
    string n4 = textBox11.Text;
    string n5 = textBox12.Text;
    string n6 = textBox13.Text;
```

Продолжение Листинг 2 – Form2.cs

```

        string n7 = textBox14.Text;
        bd.AddПрепараты(n1, n2, n3, n4, n5, n6, n7);
        textBox8.Text = "";
        textBox9.Text = "";
        textBox10.Text = "";
        textBox11.Text = "";
        textBox12.Text = "";
        textBox13.Text = "";
        textBox14.Text = "";
        bd.ShowПрепараты(dataGridView3);
    }

    private void button14_Click(object sender, EventArgs e)
    {
        string n1 = textBox8.Text;
        string n2 = textBox9.Text;
        string n3 = textBox10.Text;
        string n4 = textBox11.Text;
        string n5 = textBox12.Text;
        string n6 = textBox13.Text;
        string n7 = textBox14.Text;
        string n8 = textBox27.Text;
        bd.UpdПрепараты(n1, n2, n3, n4, n5, n6, n7, n8);
        textBox8.Text = "";
        textBox9.Text = "";
        textBox10.Text = "";
        textBox11.Text = "";
        textBox12.Text = "";
        textBox13.Text = "";
        textBox14.Text = "";
        textBox27.Text = "";
        bd.ShowПрепараты(dataGridView3);
    }

    private void button13_Click(object sender, EventArgs e)
    {
        string n8 = textBox27.Text;
        bd.DelПрепараты(n8);
        textBox27.Text = "";
        bd.ShowПрепараты(dataGridView3);
    }

    private void button19_Click(object sender, EventArgs e)
    {
        string n1 = textBox15.Text;
        string n2 = textBox16.Text;
        int n3 = Convert.ToInt32(textBox17.Text);
        int n4 = Convert.ToInt32(textBox18.Text);
        bd.AddПрепараты_аптека(n1, n2, n3, n4);
        textBox15.Text = "";
        textBox16.Text = "";
        textBox17.Text = "";
        textBox18.Text = "";
        bd.ShowПрепараты_аптека(dataGridView4);
    }

    private void button18_Click(object sender, EventArgs e)
    {
        string n1 = textBox15.Text;
        string n2 = textBox16.Text;
        int n3 = Convert.ToInt32(textBox17.Text);
        int n4 = Convert.ToInt32(textBox18.Text);
        string n5 = textBox28.Text;
        string n6 = textBox31.Text;
        bd.UpdПрепараты_аптека(n1, n2, n3, n4, n5, n6);
        textBox15.Text = "";
        textBox16.Text = "";
        textBox17.Text = "";
        textBox18.Text = "";
        textBox28.Text = "";
        bd.ShowПрепараты_аптека(dataGridView4);
    }

    private void button17_Click(object sender, EventArgs e)
    {

```

Продолжение Приложения Б

Продолжение Листинг 2 – From2.cs

```
        string n5 = textBox28.Text;
        string n6 = textBox31.Text;
        bd.DelПрепараты_аптека(n5,n6);
        textBox28.Text = "";
        bd.ShowПрепараты_аптека(dataGridView4);
    }

    private void button24_Click(object sender, EventArgs e)
    {
        string n1 = textBox19.Text;
        string n2 = textBox20.Text;
        bd.AddПрепараты_изготовитель(n1, n2);
        textBox19.Text = "";
        textBox20.Text = "";
        bd.ShowПрепараты_изготовитель(dataGridView5);
    }

    private void button23_Click(object sender, EventArgs e)
    {
        string n1 = textBox19.Text;
        string n2 = textBox20.Text;
        string n3 = textBox29.Text;
        string n4 = textBox32.Text;
        bd.UpdПрепараты_изготовитель(n1, n2, n3, n4);
        textBox19.Text = "";
        textBox20.Text = "";
        textBox29.Text = "";
        textBox32.Text = "";
        bd.ShowПрепараты_изготовитель(dataGridView5);
    }

    private void button22_Click(object sender, EventArgs e)
    {
        string n3 = textBox29.Text;
        string n4 = textBox32.Text;
        bd.DelПрепараты_изготовитель(n3, n4);
        textBox29.Text = "";
        textBox32.Text = "";
        bd.ShowПрепараты_изготовитель(dataGridView5);
    }

    private void button26_Click(object sender, EventArgs e)
    {
        SqlConnection Conn = new
SqlConnection(@"Server=localhost\SQLEXPRESS;Database=apтека;Trusted_Connection=True");
        Conn.Open();

        string countcmd = "SELECT Название FROM Препарат WHERE ID_препарата = '" + textBox30.Text +
        "','";
        SqlCommand checks = new SqlCommand(countcmd, Conn);
        var exec_check = checks.ExecuteScalar();
        string countdr2 = "SELECT Количество FROM Препарат_аптека WHERE ID_препарата = '" +
        textBox30.Text + "','";
        SqlCommand checkdr2 = new SqlCommand(countdr2, Conn);
        var exec_checkd2 = checkdr2.ExecuteScalar();
        string countdr3 = "SELECT Номер_аптеки FROM Препарат_аптека WHERE ID_препарата = '" +
        textBox30.Text + "','";
        SqlCommand checkdr3 = new SqlCommand(countdr3, Conn);
        var exec_checkd3 = checkdr3.ExecuteScalar();
        string countdr4 = "SELECT Название FROM Аптека WHERE Номер_аптеки = '" +
        exec_checkd3.ToString() + "','";
        SqlCommand checkdr4 = new SqlCommand(countdr4, Conn);
        var exec_checkd4 = checkdr4.ExecuteScalar();
        string countdr5 = "SELECT Цена FROM Препарат_аптека WHERE ID_препарата = '" + textBox30.Text +
        "','";
        SqlCommand checkdr5 = new SqlCommand(countdr5, Conn);
        var exec_checkd5 = checkdr5.ExecuteScalar();
        int a = Convert.ToInt32(exec_checkd5);
        int b = Convert.ToInt32(exec_checkd2);
        int c = a * b;
        textBox33.Text = "        Справка о наличии препарата " + exec_check.ToString() + "\n\n" + " в
        аптеке " + exec_checkd4.ToString() + "\n\n" + " Всего лекарств: " + exec_checkd2.ToString() + "\n\n" +
        " Стоимость всего реализованного: " + Convert.ToString(c) + " рублей \n\n" + " Дата: "
        + DateTime.Now.Date.ToString() + " \n\n" + "Подпись фармацевта_____ " + " \n\n" + "
```

Продолжение Приложения Б

Продолжение Листинг 2 – Form2.cs

```
Фармацевт_____";
    }

    private void printPreviewDialog1_Load(object sender, EventArgs e)
    {
    }

    private void printDocument1_PrintPage(object sender, System.Drawing.Printing.PrintPageEventArgs e)
    {
        e.Graphics.DrawString(textBox33.Text, new Font("Times New Roman", 20, FontStyle.Bold),
        Brushes.Black, new PointF(100, 100));
    }

    private void button25_Click(object sender, EventArgs e)
    {
        if (printPreviewDialog1.ShowDialog() == DialogResult.OK)
        {
            printDocument1.Print();
        }
    }
}
```

Приложение В
Листинг 3 – Form3.cs

Листинг 3 – Form3.cs

```
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 БД_аптека
{
    public partial class Form3 : Form
    {
        Dtdb bd = new Dtdb();
        public Form3()
        {
            InitializeComponent();
            bd.ShowАптека(dataGridView1);
            bd.ShowИзготовитель(dataGridView2);
            bd.ShowПрепараты(dataGridView3);
            bd.ShowПрепараты_аптека(dataGridView4);
            bd.ShowПрепараты_изготовитель(dataGridView5);
        }

        private void button7_Click(object sender, EventArgs e)
        {
            if (radioButton4.Checked == true && radioButton3.Checked == false)
            {
                bd.ShowПоиск_Изготовитель(dataGridView2, textBox26.Text);
            }
            if (radioButton3.Checked == true && radioButton4.Checked == false)
            {
                bd.ShowПоиск_Изготовитель1(dataGridView2, textBox23.Text);
            }
        }

        private void button5_Click(object sender, EventArgs e)
        {
            bd.ShowАптека(dataGridView1);
        }

        private void button6_Click(object sender, EventArgs e)
        {
            bd.ShowИзготовитель(dataGridView2);
        }

        private void button11_Click(object sender, EventArgs e)
        {
            bd.ShowПрепараты(dataGridView3);
        }
    }
}
```

Продолжение Приложения В

Продолжение Листинг 3 – Form3.cs

```
private void button30_Click(object sender, EventArgs e)
{
    bd.ShowПрепараты_аптека(dataGridView4);
}

private void button20_Click(object sender, EventArgs e)
{
    bd.ShowПрепараты_изготовитель(dataGridView5);
}

private void button4_Click(object sender, EventArgs e)
{
    if (radioButton1.Checked == true && radioButton2.Checked == false)
    {
        bd.ShowПоиск_Аптека(dataGridView1, textBox25.Text);
    }
    if (radioButton2.Checked == true && radioButton1.Checked == false)
    {
        bd.ShowПоиск_Аптека1(dataGridView1, textBox22.Text);
    }
}

private void button12_Click(object sender, EventArgs e)
{
    if (radioButton6.Checked == true && radioButton5.Checked == false)
    {
        bd.ShowПоиск_Препарат(dataGridView3, textBox27.Text);
    }
    if (radioButton5.Checked == true && radioButton6.Checked == false)
    {
        bd.ShowПоиск_Препарат1(dataGridView3, textBox24.Text);
    }
}

private void button16_Click(object sender, EventArgs e)
{
    if (radioButton8.Checked == true && radioButton7.Checked == false)
    {
        bd.ShowПоиск_Препарат_аптека(dataGridView4, textBox28.Text);
    }
    if (radioButton7.Checked == true && radioButton8.Checked == false)
    {
        bd.ShowПоиск_Препарат_аптека1(dataGridView4, textBox31.Text);
    }
}

private void button21_Click(object sender, EventArgs e)
{
    if (radioButton10.Checked == true && radioButton9.Checked == false)
    {
        bd.ShowПоиск_Препарат_изготовитель(dataGridView5, textBox29.Text);
    }
    if (radioButton9.Checked == true && radioButton10.Checked == false)
    {
        bd.ShowПоиск_Препарат_изготовитель1(dataGridView5, textBox32.Text);
    }
}
}
}
```

Приложение Г

Листинг 4 – Dtdb.cs

Листинг 4 – Dtdb.cs

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

namespace БД_аптека
{
    class Dtdb
    {
        public void ShowАптека(DataGridView data)
        {
            SqlConnection Conn = new
            SqlConnection(@"Server=localhost\SQLEXPRESS;Database=library;Trusted_Connection=True");
            Conn.Open();
            string insertcmd = "use аптека; " +
                $"select * from Аптека;";

            DataSet ds = new DataSet();

            SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);

            ds.Reset();
            da.Fill(ds);
            data.DataSource = ds.Tables[0];

            Conn.Close();
        }
        public void ShowПрепараты(DataGridView data)
        {
            SqlConnection Conn = new
            SqlConnection(@"Server=localhost\SQLEXPRESS;Database=library;Trusted_Connection=True");
            Conn.Open();
            string insertcmd = "use аптека; " +
                $"select * from Препарат;";

            DataSet ds = new DataSet();

            SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);

            ds.Reset();
            da.Fill(ds);
            data.DataSource = ds.Tables[0];

            Conn.Close();
        }
        public void ShowИзготовитель(DataGridView data)
        {
            SqlConnection Conn = new
            SqlConnection(@"Server=localhost\SQLEXPRESS;Database=library;Trusted_Connection=True");
            Conn.Open();
            string insertcmd = "use аптека; " +
                $"select * from Изготовитель;";

            DataSet ds = new DataSet();

            SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);

            ds.Reset();
            da.Fill(ds);
            data.DataSource = ds.Tables[0];

            Conn.Close();
        }
        public void ShowПрепараты_изготовитель(DataGridView data)
        {
            SqlConnection Conn = new
            SqlConnection(@"Server=localhost\SQLEXPRESS;Database=library;Trusted_Connection=True");
            Conn.Open();
            string insertcmd = "use аптека; " +
                $"select * from Препарат_изготовитель;";

            DataSet ds = new DataSet();

            SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);

            ds.Reset();
            da.Fill(ds);
            data.DataSource = ds.Tables[0];

            Conn.Close();
        }
        public void ShowПрепараты_аптека(DataGridView data)
        {
            SqlConnection Conn = new
            SqlConnection(@"Server=localhost\SQLEXPRESS;Database=library;Trusted_Connection=True");
```

Продолжение Приложения Г

Продолжение Листинг 4 – Dtdb.cs

```
        Conn.Open();
        string insertcmd = "use apteka; " +
            $"select * from Препарат_аптека;";

        DataSet ds = new DataSet();

        SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);

        ds.Reset();
        da.Fill(ds);
        data.DataSource = ds.Tables[0];

        Conn.Close();
    }
    public void ShowПоиск_Аптека(DataGridView data, string number)
    {
        SqlConnection Conn = new
SqlConnection(@"Server=localhost\SQLEXPRESS;Database=library;Trusted_Connection=True");
        Conn.Open();
        string insertcmd = "use apteka; " +
            $"select * from Аптека where Номер_аптеки = '{number}';";

        DataSet ds = new DataSet();

        SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);

        ds.Reset();
        da.Fill(ds);
        data.DataSource = ds.Tables[0];

        Conn.Close();
    }
    public void ShowПоиск_Аптека1(DataGridView data, string number)
    {
        SqlConnection Conn = new
SqlConnection(@"Server=localhost\SQLEXPRESS;Database=library;Trusted_Connection=True");
        Conn.Open();
        string insertcmd = "use apteka; " +
            $"select * from Аптека where Название = '{number}';";

        DataSet ds = new DataSet();

        SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);

        ds.Reset();
        da.Fill(ds);
        data.DataSource = ds.Tables[0];

        Conn.Close();
    }
    public void ShowПоиск_Изготовитель(DataGridView data, string number)
    {
        SqlConnection Conn = new
SqlConnection(@"Server=localhost\SQLEXPRESS;Database=library;Trusted_Connection=True");
        Conn.Open();
        string insertcmd = "use apteka; " +
            $"select * from Изготовитель where Номер_изготовителя = '{number}';";

        DataSet ds = new DataSet();

        SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);

        ds.Reset();
        da.Fill(ds);
        data.DataSource = ds.Tables[0];

        Conn.Close();
    }
    public void ShowПоиск_Изготовитель1(DataGridView data, string number)
    {
        SqlConnection Conn = new
SqlConnection(@"Server=localhost\SQLEXPRESS;Database=library;Trusted_Connection=True");
        Conn.Open();
        string insertcmd = "use apteka; " +
            $"select * from Изготовитель where Название = '{number}';";
```

Продолжение Приложения Г

Продолжение Листинг 4 – Dtdb.cs

```
        DataSet ds = new DataSet();

        SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);

        ds.Reset();
        da.Fill(ds);
        data.DataSource = ds.Tables[0];

        Conn.Close();
    }
    public void ShowПоиск_Препарат(DataGridView data, string number)
    {
        SqlConnection Conn = new
SqlConnection(@"Server=localhost\SQLEXPRESS;Database=library;Trusted_Connection=True");
        Conn.Open();
        string insertcmd = "use аптека; " +
            $"select * from Препарат where ID_препарата = '{number}';";

        DataSet ds = new DataSet();

        SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);

        ds.Reset();
        da.Fill(ds);
        data.DataSource = ds.Tables[0];

        Conn.Close();
    }
    public void ShowПоиск_Препарат1(DataGridView data, string number)
    {
        SqlConnection Conn = new
SqlConnection(@"Server=localhost\SQLEXPRESS;Database=library;Trusted_Connection=True");
        Conn.Open();
        string insertcmd = "use аптека; " +
            $"select * from Препарат where Название = '{number}';";

        DataSet ds = new DataSet();

        SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);

        ds.Reset();
        da.Fill(ds);
        data.DataSource = ds.Tables[0];

        Conn.Close();
    }
    public void ShowПоиск_Препарат_аптека(DataGridView data, string number)
    {
        SqlConnection Conn = new
SqlConnection(@"Server=localhost\SQLEXPRESS;Database=library;Trusted_Connection=True");
        Conn.Open();
        string insertcmd = "use аптека; " +
            $"select * from Препарат_аптека where ID_препарата = '{number}';";

        DataSet ds = new DataSet();

        SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);

        ds.Reset();
        da.Fill(ds);
        data.DataSource = ds.Tables[0];

        Conn.Close();
    }
    public void ShowПоиск_Препарат_аптека1(DataGridView data, string number)
    {
        SqlConnection Conn = new
SqlConnection(@"Server=localhost\SQLEXPRESS;Database=library;Trusted_Connection=True");
        Conn.Open();
        string insertcmd = "use аптека; " +
            $"select * from Препарат_аптека where Номер_аптеки = '{number}';";

        DataSet ds = new DataSet();

        SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);
```

Продолжение Приложения Г

Продолжение Листинг 4 – Dtdb.cs

```
        ds.Reset();
        da.Fill(ds);
        data.DataSource = ds.Tables[0];

        Conn.Close();
    }
    public void ShowПоиск_Препарат_изготовитель(DataGridView data, string number)
    {
        SqlConnection Conn = new
SqlConnection(@"Server=localhost\SQLEXPRESS;Database=library;Trusted_Connection=True");
        Conn.Open();
        string insertcmd = "use apteka; " +
        $"select * from Препарат_изготовитель where ID_препарата = '" + number + "';";

        DataSet ds = new DataSet();

        SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);

        ds.Reset();
        da.Fill(ds);
        data.DataSource = ds.Tables[0];

        Conn.Close();
    }
    public void ShowПоиск_Препарат_изготовитель1(DataGridView data, string number)
    {
        SqlConnection Conn = new
SqlConnection(@"Server=localhost\SQLEXPRESS;Database=library;Trusted_Connection=True");
        Conn.Open();
        string insertcmd = "use apteka; " +
        $"select * from Препарат_изготовитель where Номер_изготовителя = '" + number + "';";

        DataSet ds = new DataSet();

        SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);

        ds.Reset();
        da.Fill(ds);
        data.DataSource = ds.Tables[0];

        Conn.Close();
    }
    public void AddAnteka(string n1, string n2, string n3, string n4)
    {
        SqlConnection Conn = new
SqlConnection(@"Server=localhost\SQLEXPRESS;Database=apтека;Trusted_Connection=True");
        Conn.Open();
        string insertcmd = "INSERT INTO Anteka VALUES('" + n1 + "', '" + Convert.ToString(n2) + "', '" +
n3 + "', '" + n4 + "');";
        DataSet ds = new DataSet();
        SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);
        ds.Reset();
        da.Fill(ds);
        Conn.Close();
    }
    public void UpdAnteka(string n1, string n2, string n3, string n4, string n5)
    {
        SqlConnection Conn = new
SqlConnection(@"Server=localhost\SQLEXPRESS;Database=apтека;Trusted_Connection=True");
        Conn.Open();
        string insertcmd = "UPDATE Anteka SET Номер_аптеки = '" + n1 + "', Название = '" +
Convert.ToString(n2) + "', Адрес = '" + n3 + "', Номер_телефона = '" + n4 + "' WHERE Номер_аптеки = '" +
n5 + "';";
        SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);
        DataSet ds = new DataSet();
        ds.Reset();
        da.Fill(ds);
        Conn.Close();
    }
    public void DelAnteka(string n1)
    {
        SqlConnection Conn = new
SqlConnection(@"Server=localhost\SQLEXPRESS;Database=apтека;Trusted_Connection=True");

        Conn.Open();
        string insertcmd = "DELETE FROM Anteka WHERE Номер_аптеки = '" + n1 + "';";
```

Продолжение Приложения Г

Продолжение Листинг 4 – Dtdb.cs

```
        SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);
        DataSet ds = new DataSet();
        ds.Reset();
        da.Fill(ds);
        Conn.Close();
    }
    public void AddИзготовитель(string n1, string n2, string n3, string n4)
    {
        SqlConnection Conn = new
SqlConnection(@"Server=localhost\SQLEXPRESS;Database=apтека;Trusted_Connection=True");
        Conn.Open();
        string insertcmd = "INSERT INTO Изготовитель VALUES('" + n1 + "', '" + Convert.ToString(n2) +
        "','" + n3 + "', '" + n4 + "')";
        DataSet ds = new DataSet();
        SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);
        ds.Reset();
        da.Fill(ds);
        Conn.Close();
    }
    public void UpdИзготовитель(string n1, string n2, string n3, string n4, string n5)
    {
        SqlConnection Conn = new
SqlConnection(@"Server=localhost\SQLEXPRESS;Database=apтека;Trusted_Connection=True");
        Conn.Open();
        string insertcmd = "UPDATE Изготовитель SET Номер_изготовителя = '" + n1 + "', Название = '" +
        Convert.ToString(n2) + "', Адрес = '" + n3 + "', Номер_телефона = '" + n4 + "' WHERE Номер_аптеки = '" +
        n5 + "'";
        SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);
        DataSet ds = new DataSet();
        ds.Reset();
        da.Fill(ds);
        Conn.Close();
    }
    public void DelИзготовитель(string n1)
    {
        SqlConnection Conn = new
SqlConnection(@"Server=localhost\SQLEXPRESS;Database=apтека;Trusted_Connection=True");
        Conn.Open();
        string insertcmd = "DELETE FROM Изготовитель WHERE Номер_изготовителя = '" + n1 + "'";
        SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);
        DataSet ds = new DataSet();
        ds.Reset();
        da.Fill(ds);
        Conn.Close();
    }
    public void AddПлепараты(string n1, string n2, string n3, string n4, string n5, string n6, string
n7)
    {
        SqlConnection Conn = new
SqlConnection(@"Server=localhost\SQLEXPRESS;Database=apтека;Trusted_Connection=True");
        Conn.Open();
        string insertcmd = "INSERT INTO Пленарпат VALUES('" + Convert.ToString(n1) + "', '" +
        Convert.ToString(n2) + "', '" + Convert.ToString(n3) + "', '" + Convert.ToString(n4) + "', '" +
        Convert.ToString(n5) + "', '" + n6 + "', '" + n7 + "')";
        DataSet ds = new DataSet();
        SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);
        ds.Reset();
        da.Fill(ds);
        Conn.Close();
    }
    public void UpdПлепараты(string n1, string n2, string n3, string n4, string n5, string n6, string
n7, string n8)
    {
        SqlConnection Conn = new
SqlConnection(@"Server=localhost\SQLEXPRESS;Database=apтека;Trusted_Connection=True");
        Conn.Open();
        string insertcmd = "UPDATE Пленарпат SET ID_пленарпата = '" + Convert.ToString(n1) + "',
        Противоказания = '" + Convert.ToString(n2) + "', Название = '" + Convert.ToString(n3) + "', Вес = '" +
        Convert.ToString(n4) + "', Название_заменителя = '" + Convert.ToString(n5) + "', Срок_годности = '" + n6 +
        "', Отпуск_пореценты = '" + n7 + "' WHERE ID_пленарпата = '" + Convert.ToString(n8) + "'";
        DataSet ds = new DataSet();
        SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);
        ds.Reset();
        da.Fill(ds);
    }
```

Продолжение Приложения Г

Продолжение Листинг 4 – Dtddb.cs

```
        Conn.Close();
    }
    public void DelПенепараты(string n1)
    {
        SqlConnection Conn = new
SqlConnection(@"Server=localhost\SQLEXPRESS;Database=apтека;Trusted_Connection=True");

        Conn.Open();
        string insertcmd = "DELETE FROM Пенепарат WHERE ID_пенепарата = '" + n1 + "'";
        SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);
        DataSet ds = new DataSet();
        ds.Reset();
        da.Fill(ds);
        Conn.Close();
    }
    public void AddПенепараты_аптека(string n1, string n2, int n3, int n4)
    {
        SqlConnection Conn = new
SqlConnection(@"Server=localhost\SQLEXPRESS;Database=apтека;Trusted_Connection=True");
        Conn.Open();
        string insertcmd = "INSERT INTO Пенепарат_аптека VALUES('" + Convert.ToString(n1) + "','" +
Convert.ToString(n2) + "','" + n3 + "','" + n4 + "')";
        DataSet ds = new DataSet();
        SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);
        ds.Reset();
        da.Fill(ds);
        Conn.Close();
    }

    public void UpdПенепараты_аптека(string n1, string n2, int n3, int n4, string n5, string n6)
    {
        SqlConnection Conn = new
SqlConnection(@"Server=localhost\SQLEXPRESS;Database=apтека;Trusted_Connection=True");
        Conn.Open();
        string insertcmd = "UPDATE Пенепарат_аптека SET ID_пенепарата = '" + Convert.ToString(n1) + "',
Номер_аптеки = '" + Convert.ToString(n2) + "', Количество = " + n3 + " , Цена = " + n4 + " WHERE
ID_пенепарата = '" + Convert.ToString(n5) + "' AND Номер_аптеки = '" + Convert.ToString(n6) + "' ";
        DataSet ds = new DataSet();
        SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);
        ds.Reset();
        da.Fill(ds);
        Conn.Close();
    }
    public void DelПенепараты_аптека(string n1, string n2)
    {
        SqlConnection Conn = new
SqlConnection(@"Server=localhost\SQLEXPRESS;Database=apтека;Trusted_Connection=True");

        Conn.Open();
        string insertcmd = "DELETE FROM Пенепарат_аптека WHERE ID_пенепарата = '" + n1 + "' OR
Номер_аптеки = '" + Convert.ToString(n2) + "'";
        SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);
        DataSet ds = new DataSet();
        ds.Reset();
        da.Fill(ds);
        Conn.Close();
    }

    public void AddПенепараты_изготовитель(string n1, string n2)
    {
        SqlConnection Conn = new
SqlConnection(@"Server=localhost\SQLEXPRESS;Database=apтека;Trusted_Connection=True");
        Conn.Open();
        string insertcmd = "INSERT INTO Пенепарат_изготовитель VALUES('" + Convert.ToString(n1) + "','" +
+ Convert.ToString(n2) + "')";
        DataSet ds = new DataSet();
        SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);
        ds.Reset();
        da.Fill(ds);
        Conn.Close();
    }

    public void UpdПенепараты_изготовитель(string n1, string n2, string n3, string n4)
    {
        SqlConnection Conn = new
```

Продолжение Приложения Г

Продолжение Листинг 4 – Dtdb.cs

```
SqlConnection(@"Server=localhost\SQLEXPRESS;Database=apтека;Trusted_Connection=True");
    Conn.Open();
    string insertcmd = "UPDATE Препарат_изготовитель SET ID_препарата = '" + Convert.ToString(n1)
+ "'", Номер_изготовителя = '" + Convert.ToString(n2) + "' WHERE ID_препарата = '" + Convert.ToString(n3) +
+ "' AND Номер_изготовителя = '" + Convert.ToString(n4) + "'";
    DataSet ds = new DataSet();
    SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);
    ds.Reset();
    da.Fill(ds);
    Conn.Close();
}
public void DelПрепараты_изготовитель(string n1, string n2)
{
    SqlConnection Conn = new
SqlConnection(@"Server=localhost\SQLEXPRESS;Database=apтека;Trusted_Connection=True");

    Conn.Open();
    string insertcmd = "DELETE FROM Препарат_изготовитель WHERE ID_препарата = '" + n1 + "' OR
Номер_изготовителя = '" + Convert.ToString(n2) + "'";
    SqlDataAdapter da = new SqlDataAdapter(insertcmd, Conn);
    DataSet ds = new DataSet();
    ds.Reset();
    da.Fill(ds);
    Conn.Close();
}
}
```