СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc161605318)

[1 ОБЕСПЕЧЕНИЕ КАЧЕСТВА И СЕРТИФИКАЦИЯ ИНФОРМАЦИОННЫХ СИСТЕМ 5](#_Toc161605319)

[1.1 Анализ нормативно-правовой документации, стандартов, регламентирующие качество программных средств 5](#_Toc161605320)

[1.1.1 Факторы качества программного обеспечения 5](#_Toc161605321)

[1.1.2 Законодательство Российской Федерации в области защиты информации 5](#_Toc161605322)

[1.2 Системы сертификации. Процедура сертификации 7](#_Toc161605323)

[1.2.1 Основные определения в области сертификации и лицензирования деятельности в сфере информатизации в Российской Федерации 7](#_Toc161605324)

[1.2.2 Сертификация средств и систем информатизации в Российской Федерации 8](#_Toc161605325)

[1.2.3 Лицензирование деятельности в сфере информатизации в Российской Федерации 10](#_Toc161605326)

[2 ТЕХНОЛОГИИ АДМИНИСТРИРОВАНИЯ СЕРВЕРОВ И БАЗ ДАННЫХ 12](#_Toc161605327)

[2.1 Проектирование базы данных 12](#_Toc161605328)

[2.1.1 Описание предметной области и основных требований, предъявляемых к информационной системе 12](#_Toc161605329)

[2.1.2 Описание групп пользователей информационной системы 13](#_Toc161605330)

[2.1.3 Моделирование диаграммы вариантов использования 13](#_Toc161605331)

[2.1.4 Моделирование данных, построение диаграммы «сущность-связь» 14](#_Toc161605332)

[2.2 Разработка и администрирование базы данных 17](#_Toc161605333)

[2.2.1 Создание физической модели данных 17](#_Toc161605334)

[2.2.2 Разработка серверной части базы данных в инструментальной среде 18](#_Toc161605335)

[2.2.3 Построение запросов к базе данных на языке SQL. 21](#_Toc161605336)

[2.3 Разработка клиентской части приложения в инструментальной среде 25](#_Toc161605337)

[2.4 Защита данных в хранилищах 41](#_Toc161605338)

[2.4.1 Определение привилегий пользователей с различными ролями. 41](#_Toc161605339)

[2.4.2 Настройка прав доступа к базе данных. 44](#_Toc161605340)

[2.4.3 Создание резервных копий базы данных. Восстановление базы данных из резервной копии 45](#_Toc161605341)

[ЗАКЛЮЧЕНИЕ 47](#_Toc161605342)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 48](#_Toc161605343)

ВВЕДЕНИЕ

Практическая подготовка в виде учебной практики необходима для демонстрации полученных навыков и степени освоения знаний, полученных в процессе лекционных, лабораторных и самостоятельных занятий.

В процессе освоения модуля, обучающиеся овладевают следующими видами профессиональных компетенций:

ПК 7.1. Выявлять технические проблемы, возникающие в процессе эксплуатации баз данных и серверов.

ПК 7.2. Осуществлять администрирование отдельных компонент серверов.

ПК 7.3. Формировать требования к конфигурации локальных компьютерных сетей и серверного оборудования, необходимые для работы баз данных и серверов.

ПК 7.4. Осуществлять администрирование баз данных в рамках своей компетенции.

ПК 7.5. Проводить аудит систем безопасности баз данных и серверов, с использованием регламентов по защите информации.

Разработка автоматизированной системы обработки данных о клиентах и заказах представляет собой актуальную и востребованную задачу в современном бизнесе. В рамках практической подготовки в виде учебной практики будут приобретены практические навыки в области анализа соадминистрирования баз данных и серверов.

Целью учебной практики является применение навыков разработки и внедрения автоматизированной системы заказов, способной оптимизировать процессы управления заказами через каталог услуг и повысить эффективность строительной фирме.

Для достижения поставленной цели и реализации задач учебной практики будет проводиться анализ предметной области, проектирование архитектуры системы, реализация функционала и тестирование разработанного программного обеспечения.

1 ОБЕСПЕЧЕНИЕ КАЧЕСТВА И СЕРТИФИКАЦИЯ ИНФОРМАЦИОННЫХ СИСТЕМ

1.1 Анализ нормативно-правовой документации, стандартов, регламентирующие качество программных средств

Для обеспечения качества программных средств существует ряд нормативно-правовых документов, стандартов и регламентов. В этом разделе описываются все международные и российские стандарты, описывающие и регламентирующие качество программной продукции и программных средств. Кроме того, существуют также отраслевые стандарты и регламенты, разработанные организациями и сообществами специалистов в области программной инженерии. Важно следить за изменениями в законодательстве и стандартах, чтобы быть в курсе требований качества программного обеспечения и соответствовать им.

1.1.1 Факторы качества программного обеспечения

Международный стандарт ISO 9126:1-4:2002 международный стандарт, определяющий оценочные характеристики качества программного обеспечения. Российский аналог стандарта ГОСТ Р ИСО/МЭК 9126-93. В 2011 году был заменён стандартом ISO/IEC 25010:2011. Состоит из 4 частей, описывающие следующие аспекты: модель качества; внешние метрики качества; внутренние метрики качества; метрики качества в использовании. Он описывает множество характеристик качества программного обеспечения, в отчете представлены шесть основных характеристик:

1. Функциональность (Functionality): Включает в себя совокупность атрибутов, позволяющих программе удовлетворять заявленные и обнаруженные потребности. Это включает в себя атрибуты, такие как правильность, пригодность, взаимозаменяемость, соответствие, безопасность.

2. Надежность (Reliability): Эта характеристика отражает способность программного обеспечения поддерживать заявленные функции при заданных условиях в течение определенного периода времени. К ней относятся атрибуты, такие как устойчивость к отказам, восстановление после сбоев, точность, стабильность.

3. Удобство использования (Usability): Эта характеристика оценивает удобство программы для пользователей, включая атрибуты, такие как понятность, обучаемость, удобство использования, внешний вид.

4. Эффективность (Efficiency): Оценивает способность программы выполнять свои функции при заданных условиях, относясь к атрибутам, таким как время ответа, использование ресурсов, производительность.

5. Поддерживаемость (Maintainability): Характеристика, отражающая способность программного обеспечения быть модифицированным и обеспечивать его сопровождение. Включает в себя атрибуты, такие как анализируемость, изменяемость, устойчивость ко внештатным ситуациям.

6. Переносимость (Portability): Эта характеристика оценивает способность программы быть установленной и выполненной в различных окружениях. К ней относятся атрибуты, такие как адаптивность, удобство модификации, переносимость к платформам.

1.1.2 Законодательство Российской Федерации в области защиты информации

Проанализировав актуальные (действующие) варианты законов Российской федерации в области защиты информации, приведены аннотации к закону представленные в таблице №1.

Таблица 1 – Законы РФ в области защиты информации

| Название | Описание |
| --- | --- |
| Федеральный закон от 28 декабря 2010 г. N 390-ФЗ "О безопасности" (изм. от 10.07.2023 N 286-ФЗ) | Уточняются принципы и содержание деятельности по обеспечению безопасности. Закрепляются основные цели международного сотрудничества в данной области. |
| Федеральный закон от 18.12.2006 N 231-ФЗ (ред. от 29.12.2022) "О введении в действие части четвертой Гражданского кодекса Российской Федерации" | "Основы государственного регулирования торговой деятельности в Российской Федерации", имеет целью установление основ государственного регулирования торговой деятельности в Российской Федерации, создание правовых условий для развития торговли и защиты прав потребителей.  Основные назначения данного закона включают:  1. Установление правовых основ для проведения торговой деятельности в Российской Федерации, включая рынки, ярмарки, аукционы и другие формы торговли.  2. Регулирование вопросов аренды торговых площадей и правил их использования.  3. Защита конкуренции и предотвращение злоупотреблений на рынке.  4. Создание условий для защиты прав потребителей и прозрачности торговой деятельности.  5. Установление требований к торговому оборудованию и технологиям торговли.  6. Регулирование вопросов, связанных с продажей алкогольной и табачной продукции.  7. Установление порядка проведения торгов.  8. Определение правил проведения розничной торговли, включая требования к организации и оборудованию торговых объектов. |
| "Гражданский кодекс Российской Федерации (часть четвертая)" от 18.12.2006 N 230-ФЗ (ред. от 01.07.2017) (с изм. и доп., вступ. в силу с 01.01.2018) | Переход исключительного права на зарегистрированные программу для ЭВМ или базу данных к другому лицу, по договору, или без договора и залог исключительного права на зарегистрированные программу для ЭВМ или базу данных, подлежат государственной регистрации в федеральном органе исполнительной власти по интеллектуальной собственности. |
| Федеральный закон от 20.07.2004 N 72-ФЗ "О внесении изменений в Закон Российской Федерации "Об авторском праве и смежных правах" | Закон утратил силу в связи с поправкой в гражданский кодекс российской федерации и добавлением в него 4 статьи |
| ЗАКОН РФ от 21.07.1993 N 5485-1 "О ГОСУДАРСТВЕННОЙ ТАЙНЕ" Редакция от 04.08.2023 (с изм. и доп., вступ. в силу с 01.02.2024) | Закон РФ "О государственной тайне" (редакция от 04.08.2023) имеет целью установление правовых основ в области государственной тайны. Назначение данного закона заключается в регулировании вопросов, связанных с определением, классификацией, хранением, использованием и раскрытием сведений, составляющих государственную тайну.  Основные назначения данного закона включают:  1. Установление порядка и правил определения информации в качестве государственной тайны и ее классификации в соответствии с уровнями секретности.  2. Установление прав и обязанностей субъектов, имеющих доступ к государственной тайне, в том числе по вопросам использования, хранения и раскрытия такой информации.  3. Обеспечение конфиденциальности сведений, имеющих особое значение для обеспечения безопасности государства.  4. Установление правил и процедур для контроля за соблюдением режима государственной тайны.  5. Установление ответственности за неправомерное раскрытие сведений, составляющих государственную тайну. |
| Федеральный закон от 01.05.2022 N 77-ФЗ "ОБ ОБЯЗАТЕЛЬНОМ ЭКЗЕМПЛЯРЕ ДОКУМЕНТОВ" | Производители документов в целях последующего распределения изданий между крупнейшими библиотечно-информационными организациями доставляют в Российскую государственную библиотеку обязательные федеральные экземпляры всех видов печатных изданий. |
| Федеральный закон от 06.04.2011 N 63-ФЗ (ред. от 04.08.2023) "Об электронной подписи" (с изм. и доп., вступ. в силу с 01.09.2023) | Федеральный закон "Об электронной подписи" (ред. от 04.08.2023) регулирует использование и применение электронной подписи в Российской Федерации. Назначение данного закона заключается в установлении правовых основ для использования электронной подписи с целью достижения юридической значимости электронных документов и обеспечения безопасности электронного документооборота.  Основные назначения данного закона включают:  1. Установление правовых основ для создания, проверки, использования и хранения электронной подписи, направленных на подтверждение авторства документа и целостности информации.  2. Регулирование отношений, возникающих в процессе использования электронной подписи, в том числе при заключении и исполнении договоров с использованием электронных документов.  3. Установление требований к квалифицированным услугам по работе с электронной подписью и квалифицированным удостоверяющим центрам.  4. Создание условий для использования электронной подписи в различных сферах деятельности. |
| Федеральный закон от 29.07.2004 N 98-ФЗ (ред. от 14.07.2022) "О коммерческой тайне" | Федеральный закон "О коммерческой тайне" (в редакции от 14.07.2022) имеет целью установление правовых основ для охраны информации, которая составляет коммерческую тайну, то есть информации, подлежащей сохранению в тайне в интересах обеспечения экономической деятельности и недопущения недобросовестной конкуренции. |
| Федеральный закон от 27.07.2006 N 149-ФЗ (ред. от 12.12.2023) "Об информации, информационных технологиях и о защите информации" | Федеральный закон от "Об информации, информационных технологиях и о защите информации" имеет целью установление правовых основ в области информации, информационных технологий и защиты информации в Российской Федерации. Основной задачей данного закона является обеспечение безопасности информационных ресурсов, защита конфиденциальности и целостности информации, а также обеспечение соблюдения законодательства в области информационных технологий. |

1.2 Системы сертификации. Процедура сертификации

1.2.1 Основные определения в области сертификации и лицензирования деятельности в сфере информатизации в Российской Федерации

* Сертификация: Сертификация в области информатизации в Российской Федерации - это процедура подтверждения соответствия продукции, услуг, систем управления или процессов определенным стандартам, нормативам или техническим условиям. Сертификация проводится аккредитованными организациями и может включать в себя процессы испытаний, проверки и аудита.
* Лицензирование деятельности в сфере информатизации: Лицензирование в области информатизации в Российской Федерации представляет собой процедуру выдачи правительственного разрешения (лицензии) на осуществление определенных видов деятельности, связанных с информационными технологиями. Цель лицензирования - обеспечение контроля за качеством и безопасностью информационных технологий, защита прав и интересов пользователей.
* Аккредитованная организация: Аккредитованная организация в области информатизации в Российской Федерации - это организация, которая имеет официальное признание и разрешение на проведение сертификации продукции, услуг или систем управления, в соответствии с требованиями стандартов и нормативов.
* Информационные технологии (ИТ): Под информационными технологиями понимается совокупность средств и методов обработки, хранения, передачи и использования информации с использованием компьютеров и программного обеспечения.
* Безопасность информации: Безопасность информации включает в себя защиту информации от несанкционированного доступа, утечки, порчи и других угроз, а также обеспечение целостности, конфиденциальности и доступности информации. В рамках информатизации, это одно из ключевых понятий, которое регулируется лицензионными требованиями и стандартами сертификации.

1.2.2 Сертификация средств и систем информатизации в Российской Федерации

Организация работ по сертификации средств и систем информатизации в Российской Федерации включает важные этапы, связанные с подготовкой, проведением и завершением процедуры сертификации. Ниже представлены основные этапы организации работ по сертификации средств и систем информатизации:

1. Подготовка к сертификации:

* определение категории продукции или услуги, требующей сертификации в соответствии с законодательством Российской Федерации;
* выбор аккредитованной организации, которая будет осуществлять сертификационные испытания и проверку соответствия;
* подготовка необходимой документации в соответствии с требованиями законодательства и стандартов;

2. Проведение сертификационных испытаний:

* подписание договора с аккредитованной организацией о проведении испытаний;
* проведение необходимых испытаний продукции или системы, включая проверку соответствия техническим требованиям и стандартам;
* предоставление образцов продукции или доступ к системе для проведения испытаний.

3. Оценка соответствия:

* анализ результатов сертификационных испытаний и подготовка заключения об их соответствии или несоответствии установленным требованиям;
* подготовка отчета о проведенных испытаниях и анализе результатов.

4. Выдача сертификата:

* при положительных результатах испытаний выдается сертификат о соответствии продукции или системы установленным стандартам и требованиям;
* регистрация сертификата в установленном порядке и внесение в реестр сертифицированных продуктов или систем.

5. Контроль качества:

* Организация контроля за качеством сертифицированной продукции или системы в соответствии с законодательством и требованиями стандартов.

Эти этапы представляют общий контекст организации работ по сертификации средств и систем информатизации в Российской Федерации, но в каждом конкретном случае могут быть уточнения и дополнительные требования, зависящие от конкретной продукции или системы.

1.2.3 Лицензирование деятельности в сфере информатизации в Российской Федерации

Организация работ по лицензированию деятельности в сфере информатизации в Российской Федерации осуществляется в соответствии с законодательством и общими принципами, установленными в этой области. Вот основные общие принципы организации работ по лицензированию в сфере информатизации:

* определение видов деятельности: На первом этапе необходимо определить виды деятельности в сфере информатизации, которые подлежат обязательному лицензированию. Это могут быть, например, деятельность по разработке и внедрению программного обеспечения, работы связанные с обработкой персональных данных, информационно-технологическое сопровождение и т.д.;
* сбор необходимой документации: Лицензирование включает в себя предоставление определенного перечня документов, подтверждающих соответствие заявителя установленным требованиям. Это может включать в себя регистрацию юридического лица, наличие специальных образовательных или профессиональных квалификаций, наличие определенных финансовых средств и так далее;
* заявление и подача документов: Заявитель обращается в специально уполномоченный орган с заявлением о лицензировании, представляет необходимые документы и информацию;
* экспертиза документов: Уполномоченный орган проводит экспертизу предоставленных документов, проверяет их наличие и соответствие установленным требованиям;
* проведение проверок: В некоторых случаях лицензирующий орган может проводить проверки на предприятии или в офисе заявителя для убедительности в предоставленных данных;
* выдача лицензии: При положительном результате рассмотрения заявления и предоставленной документации уполномоченный орган выдает лицензию на осуществление конкретных видов деятельности в сфере информатизации;
* соблюдение лицензионных требований: Владельцы лицензий обязаны соблюдать все установленные лицензионные требования, в противном случае им может быть отказано в продлении лицензии или она может быть аннулирована.

Эти общие принципы являются основой для организации работ по лицензированию деятельности в сфере информатизации в Российской Федерации. Однако следует отметить, что конкретные процедуры и требования могут различаться в зависимости от каждого конкретного вида деятельности и от специфики регулирующих органов.

Вывод

По окончанию первой главы был проведен анализ предметной области. Описаны основные факторы качества программного обеспечения, сертификации средств и систем информатизации в Российской Федерации, описаны Законы РФ в области защиты информации и их краткое описание, этапы сертификации программного продукта.

2 ТЕХНОЛОГИИ АДМИНИСТРИРОВАНИЯ СЕРВЕРОВ И БАЗ ДАННЫХ

2.1 Проектирование базы данных

В данной главе описывается предметная область, требования, предъявляемые к ней моделирование диаграмм и разработка пользовательского интерфейса для работы с базой данных.

2.1.1 Описание предметной области и основных требований, предъявляемых к информационной системе

Информационная система "Система учета заказов и их выполнение в строительной фирме" представляет собой программно-аппаратный комплекс, предназначенный для управления, автоматизации и оптимизации процесса заказов услуг. Основной целью данной информационной системы является упрощение процесса формирования заказов для клиентов, сокращение времени на обработку и повышение эффективности работы персонала компании.

Система должна позволять обрабатывать данных о клиентах и перечне услуг, а также учете заказов, используемом материале и учете затрат по заказам. Вместе с этим в системе должна храниться база фотографий с образцами ремонта и в целом отремонтированных квартир.

Основные требования к информационной системе "Система учета заказов и их выполнение в строительной фирме " могут включать следующие аспекты:

* Управление каталогом услуг: Система должна предоставлять возможность управления ассортиментом услуг в каталоге, включая добавление новых услуг, редактирование существующих и удаление устаревших.
* Безопасность: Система должна обеспечивать защиту информации о клиентах и заказах, включая применение мер безопасности для предотвращения несанкционированного доступа и сохранения конфиденциальности данных.
* Интерфейс пользователя: Дружественный пользовательский интерфейс, который обеспечивает простоту использования и понимания процесса оформления заказа. Интерфейс должен быть интуитивно понятным и удобным для пользователя.
* Масштабируемость: Система должна быть гибкой и масштабируемой, способной адаптироваться к изменяющемуся объему заказов и росту бизнеса.

Дополнительные требования могут зависеть от специфики бизнес-процессов организации, таких как специальные требования по интеграции с внешними системами, расширенные возможности аналитики и отчетности, мобильная совместимость и т.д.

2.1.2 Описание групп пользователей информационной системы

Здесь представлено объяснение действующих лиц и функционал каждого из них, какие действия они могут выполнять:

1. Администраторы системы:

* осуществляют управление пользователями системы, их правами и доступом к функционалу;
* отвечают за настройку и обслуживание системы, обновление каталога услуг, добавление новых пользователей и управление правами доступа.

2. Менеджеры:

* осуществляют управление заказами, обработку заказов клиентов, управление каталогом услуг и ценовой политикой;
* могут генерировать отчеты о продажах, аналитическую информацию и статистику по заказам.

3. Директор:

* используют систему для просмотра каталога.

2.1.3 Моделирование диаграммы вариантов использования

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью, так называемых, вариантов использования. При этом актером или действующим лицом называется любая сущность, взаимодействующая с системой извне. Это может быть человек, техническое устройство, программа или любая другая система, которая может служить источником воздействия на моделируемую систему так как определит сам разработчик. Диаграмма вариантов использования представлена на рисунке 1.

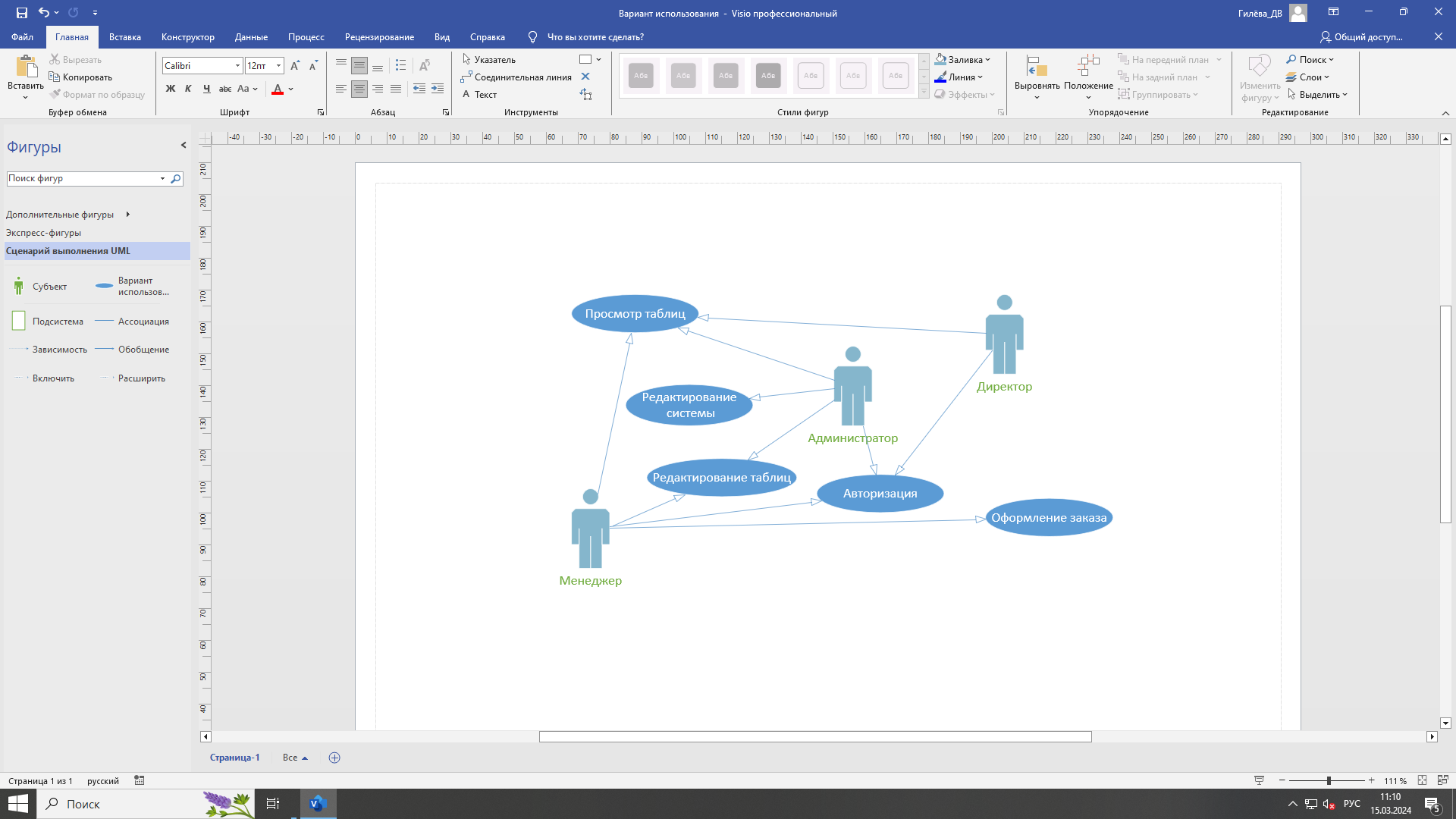


Рисунок 1 – Диаграмма вариантов использования

В базе данных необходимо предоставить пользователю имена для входа с определенными правами:

* Менеджер –доступ к просмотру и редактированию всех таблиц;
* Директор – просмотр основных таблиц, без возможности их редактирования и возможности чтения никаких других таблиц;
* Администратор – привилегия SELECT на все таблицы базы данных и привилегия INSERT на основные таблицы.

2.1.4 Моделирование данных, построение диаграммы «сущность-связь»

Необходимо создать базу данных, отражающую процесс заказа услуг Клиентом по каталогу.

Исходя из анализа предметной области, можно выделить 5 типов сущности (таблиц): Заказ, Услуги, Роли, Пользователи, Учет материалов, которые связаны между собой типом отношений (Много-ко-многим), так как каждый покупатель может купить много наименований услуг, а каждая услуга может быть куплена многими покупателями. Однако, реляционная модель данных требует заменить отношение Многие-ко-многим на несколько отношений Один-ко-многим, для этого необходимо в структуру бедующей базы данных добавить ещё одну сущность (таблицу) Корзина услуг.

**Описание отношений:**

* одному заказу, может соответствовать только одна корзина услуг, поэтому между объектами Услуги и Заказ имеется связь Один-к-одному;
* одной роли может соответствовать несколько пользователей, поэтому между объектами Роли и Пользователи имеется связь Один-ко-многим;
* один пользователь может иметь несколько заказов, поэтому между объектами Пользователь и Заказ имеется связь Один-ко-многим;
* в одной корзине может находиться несколько услуг, поэтому между объектами Корзина услуг и Услуги имеется связь Один-ко-многим.
* У одной услуги может быть несколько материалов, поэтому между объектами Услуги и Учет услуг имеется связь Один-ко-многим;

**Описание сущностей:**

* к объекту Учет материалов относятся такие характеристики как: Наименование, Код услуги, Цена, Кол-во;
* к объекту Корзина услуг относится: Код Корзины, Код услуги;
* к объекту Заказ относится: Код Корзины, Код Пользователя, Статус;
* к объекту Услуги относятся: Наименование, Стоимость;
* к объекту Роль относятся: Наименование;
* к объекту Пользователь относятся: Имя, Фамилия, Отчество, Логин, Пароль, Код Роли.

**Описание индексов:**

В таблице Услуги необходимо определить первичный ключ - Код услуги, под которым можно понимать номер или артикул заказа. В составной индекс таблицы Услуги (поля внешнего ключа) входит поле: Код Услуги.

В таблице Корзина услуг в составной индекс таблицы Корзина (поля внешнего ключа) входит поле: Код Услуги, Код Корзины.

В таблице Заказ первичным ключом будет поле - Код Заказа, так как оно однозначно определяет Заказ, сделанный определённым пользователем. В составной индекс таблицы Заказ (поля внешнего ключа) входят поля: Код Корзины, Код Пользователя.

В таблице Роль первичным ключом будет поле - Код Роли, так как оно однозначно определяет Роль по её серийному номеру или коду.

В таблице Учет материалов первичным ключом будет поле - Код материала, так как оно однозначно определяет Номер материала для лучшего определения.

В таблице Пользователь первичным ключом будет поле - Код Пользователя, так как оно однозначно определяет пользователя по его серийному номеру (номер телефона, серия и номер паспорта и т.д.). В составной индекс таблицы Пользователи (поля внешнего ключа) входит поле: Код роли.

**Описание связей:**

* для обеспечения целостности данных между таблицами Корзина Услуг и Услуги по полю Код Услуги тип связи определен как Один-ко-многим;
* между таблицами Учет материалов и Услуги, по полю Код Услуги тип связи определен как Один-ко-многим;
* между таблицами Пользователь и Роль по полю Код роли тип связи установлен как Один-ко-многим;
* между таблицами пользователь и Заказ по полю Код пользователя тип связи установлен как Один-ко-многим;
* между таблицами Корзина услуг и Заказ по полю Код Корзины тип связи установлен как Один-к-одному.

Логическая модель данных базы данных представлена на рисунке 2.

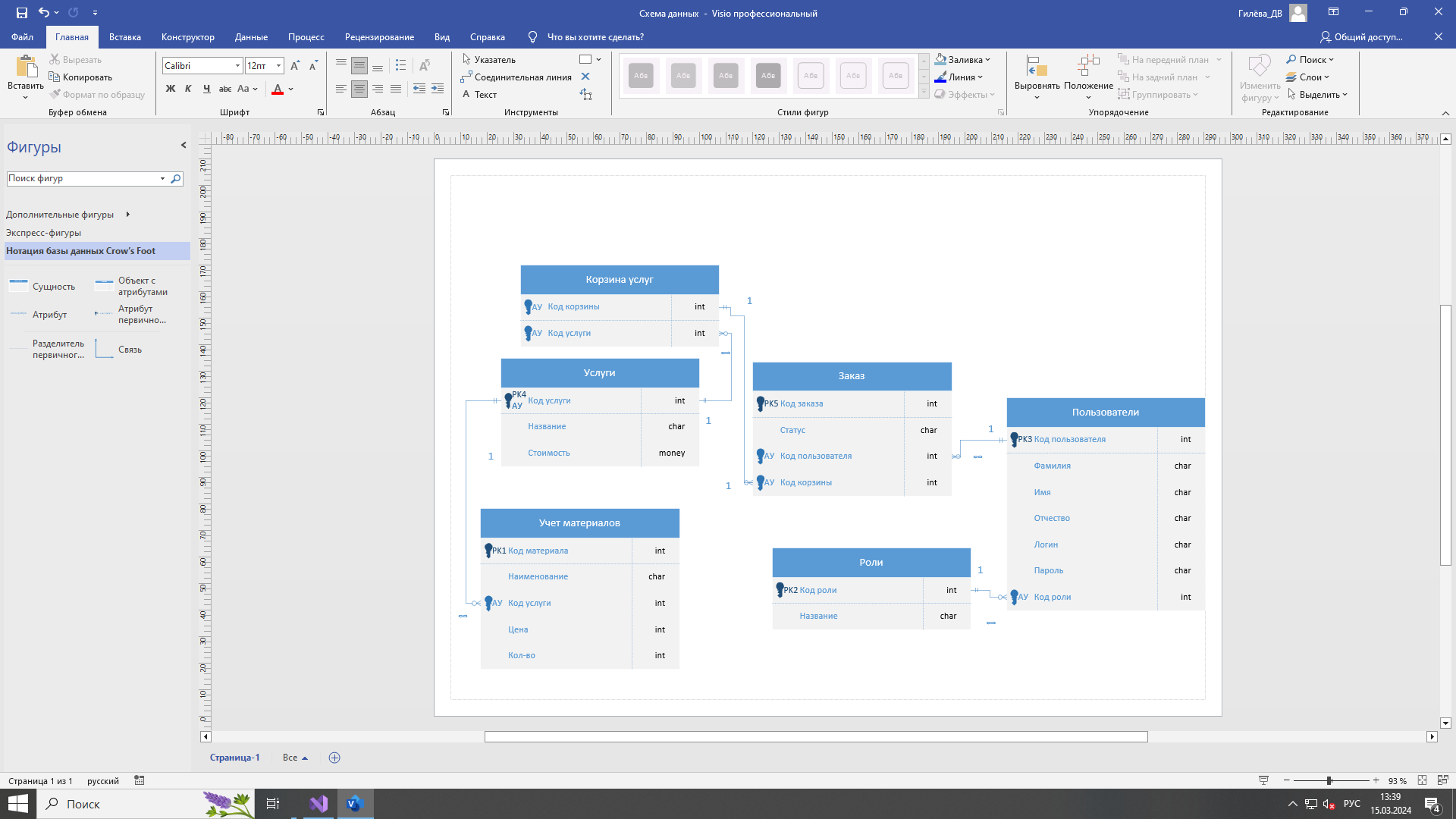


Рисунок 2 - Логическая модель данных

2.2 Разработка и администрирование базы данных

2.2.1 Создание физической модели данных

Используя систему управления базами данных SQL Server и неструктурированный язык запросов Transact –SQL с помощью SQL кода создана база данных.

В ходе выполнения заданий практической подготовки в виде учебной практики велась статистика состояния базы данных в процессе ее изменения. Результаты представлены в таблице 2.

Таблица 2 – Статистика состояния базы данных

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Название этапа | Размер базы | Нераспределенное пространство | Зарезервировано | Данные | Размер индекса | Не использовано |
| Создание БД | 400.00 MB | 197.60 MB | 2456 KB | 992 KB | 1072 KB | 392 KB |
| Создание таблиц | 400.00 MB | 197.60 MB | 2648KB | 1048KB | 1096KB | 504KB |
| Заполнение таблиц данными | 400.00 MB | 196.92 MB | 3152KB | 1112KB | 1144KB | 896KB |
| Создание пользователей | 400.00 MB | 196.38 MB | 3712KB | 1224KB | 1256KB | 1232KB |
| Формирование запросов | 400.00 MB | 196.38 MB | 3712KB | 1224KB | 1256KB | 1232KB |

2.2.2 Разработка серверной части базы данных в инструментальной среде

Используя систему управления базами данных SQL Server и неструктурированный язык запросов Transact –SQL с помощью SQL кода создавались таблицы базы данных в соответствии с построенной в п.2.1.4 логической моделью данных (PK и FK создавались в одном запросе).

Используя систему управления базами данных SQL Server, с помощью визуальных компонентов, создана Диаграмма схемы данных представленная на рисунке 3.

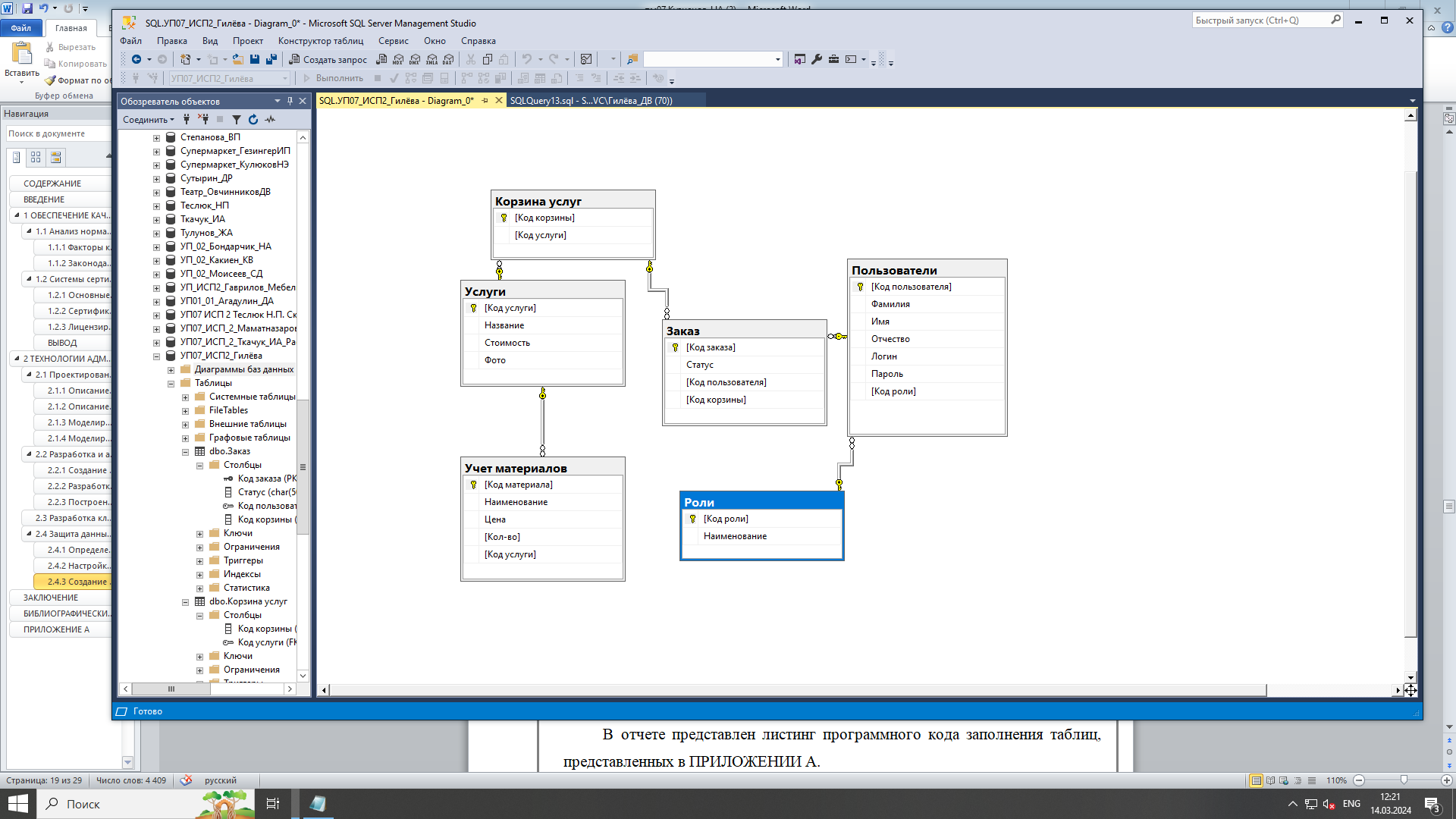


Рисунок 3 – Диаграмма схемы данных в SQL Server

Используя систему управления базами данных SQL Server и неструктурированный язык запросов Transact – SQL с помощью SQL кода заполнены таблицы корректными данными. Количество данных в таблицах обеспечивает выдачу не менее 3-5 записей по каждому запросу задания.

В отчете представлен листинг программного кода создания и заполнения таблиц, представленных ниже:

Создание БД

CREATE DATABASE УП07\_ИСП2\_Гилёва

ON PRIMARY (

NAME=УП07\_ИСП2Гилёва,

FILENAME='\\sql\c$\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\УП07\_ИСП2Гилёва.mdf',

SIZE=200MB,

MAXSIZE=300, FILEGROWTH=20)

LOG ON (

NAME= archlog\_ГилёваДВ,

FILENAME='\\sql\c$\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\archlog\_ГилёваДВ.ldf',

SIZE=200MB,

MAXSIZE=300,

FILEGROWTH=20

)

Создание таблиц:

Листинг программного кода создания таблицы Пользователи:

CREATE TABLE Пользователи

( [Код пользователя] int identity(1,1) constraint pk3 PRIMARY KEY ([Код пользователя]),

Фамилия CHAR (50) NOT NULL,

Имя CHAR (50) NOT NULL,

Отчество CHAR (50) NOT NULL,

Логин CHAR (50) NOT NULL,

Пароль CHAR (50) NOT NULL,

[Код роли]int CONSTRAINT fk2 FOREIGN KEY ([Код роли]) REFERENCES Роли ([Код роли]))

Листинг программного кода создания таблицы Роли:

CREATE TABLE Роли

( [Код роли] int identity(1,1) constraint pk2 PRIMARY KEY ([Код роли]),

Наименование CHAR (50) NOT NULL)

Листинг программного кода создание таблицы Учет материалов:

CREATE TABLE [Учет материалов]

( [Код материала] int identity(1,1) constraint pk1 PRIMARY KEY ([Код материала]),

Наименование CHAR (50) NOT NULL,

Цена INT NOT NULL,

[Кол-во] INT NOT NULL,

[Код услуги]int CONSTRAINT fk4 FOREIGN KEY ([Код услуги]) REFERENCES Услуги ([Код услуги]))

Листинг программного кода создание таблицы Услуги:

CREATE TABLE Услуги

( [Код услуги] int identity(1,1) constraint pk4 PRIMARY KEY ([Код услуги]),

Название CHAR (50) NOT NULL,

Стоимость MONEY NOT NULL,

Фото IMAGE)

Заполнение таблицы Роль данными:

insert into Роли

Values ('администратор'),

('директор'),

('менеджер')

Заполнение таблицы Пользователи данными:

insert into Пользователи

Values ('Будкин','Александр','Александрович','nikih1232','123123','1'),

('Таранин','Владимир','Алексеевич','azilka01','12345','2'),

('Гилёва','Диана','Владимировна','memekys3000','12345','3'),

('Телегин','Никита','Вячеславович','Lol268','352473','1'),

('Грибоедов','Артур','Максимович','grib5555','5463','1')

Заполнение таблицы Услуги данными:

insert into Услуги

values ('Дизайн проектов','20000',),

('укрепленность помещений','15000',),

('Демонтаж','10000',),

('Ремонт помещений','90000',),

('Отделка помещений','60000',),

('Монтаж','35000',)

Заполнение таблицы Учет материалов данными:

insert into [Учет материалов]

values ('Цемент','2000','45','1'),

('Бетон','2500','48','2'),

('Кирпич','4000','11','3'),

('Стекло','5500','22','4'),

('Известь','7500','45','5'),

('Пластик','5000','54','6')

Заполнение таблицы Заказ данными:

insert into Заказ

values ('Обрабатывается','9','1'),

('Выполнен','10','2'),

('Выполнен','11','3'),

('Обрабатывается','12','4'),

('Обрабатывается','13','5'),

('Выполнен','14','6')

Заполнение таблицы Корзина услуг данными:

insert into [Корзина услуг]

Values ('1','2'),

('2','3'),

('6','4')

2.2.3 Построение запросов к базе данных на языке SQL.

Используя неструктурированный язык запросов Transact –SQL с помощью SQL кода создать запросы на выборку данных из таблиц базы данных:

1. Выборка данных из связанных таблиц.

Составить список кто, какую услугу, с выводом общей суммы покупки. Листинг программного кода представлен ниже, Предполагаемый план выполнения представлен на рисунке 4, результат выполнения – на рисунке 5.

CREATE VIEW запрос1 as

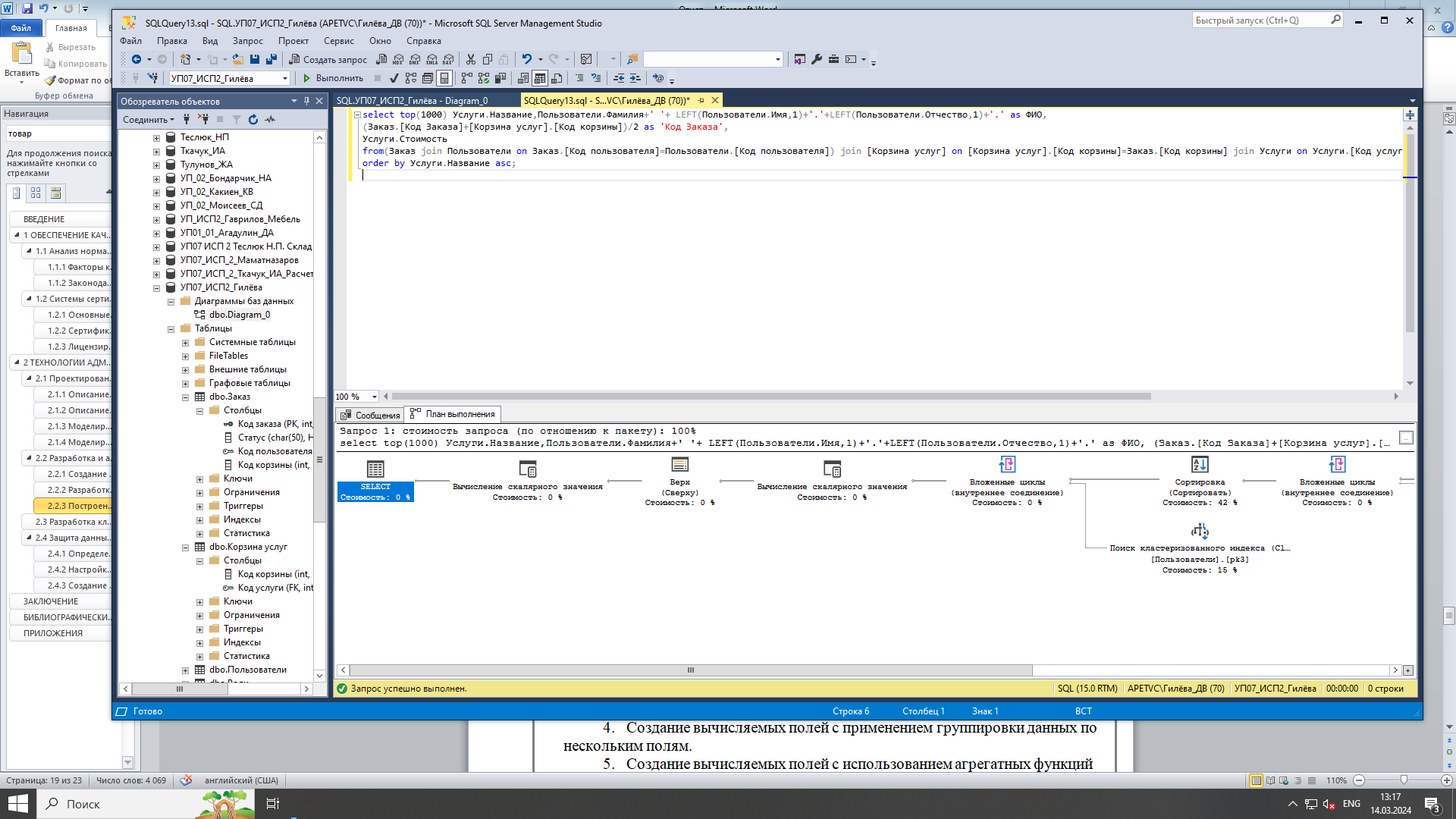
select top(1000) Услуги.Название,Пользователи.Фамилия+' '+ LEFT(Пользователи.Имя,1)+'.'+LEFT(Пользователи.Отчество,1)+'.' as ФИО,

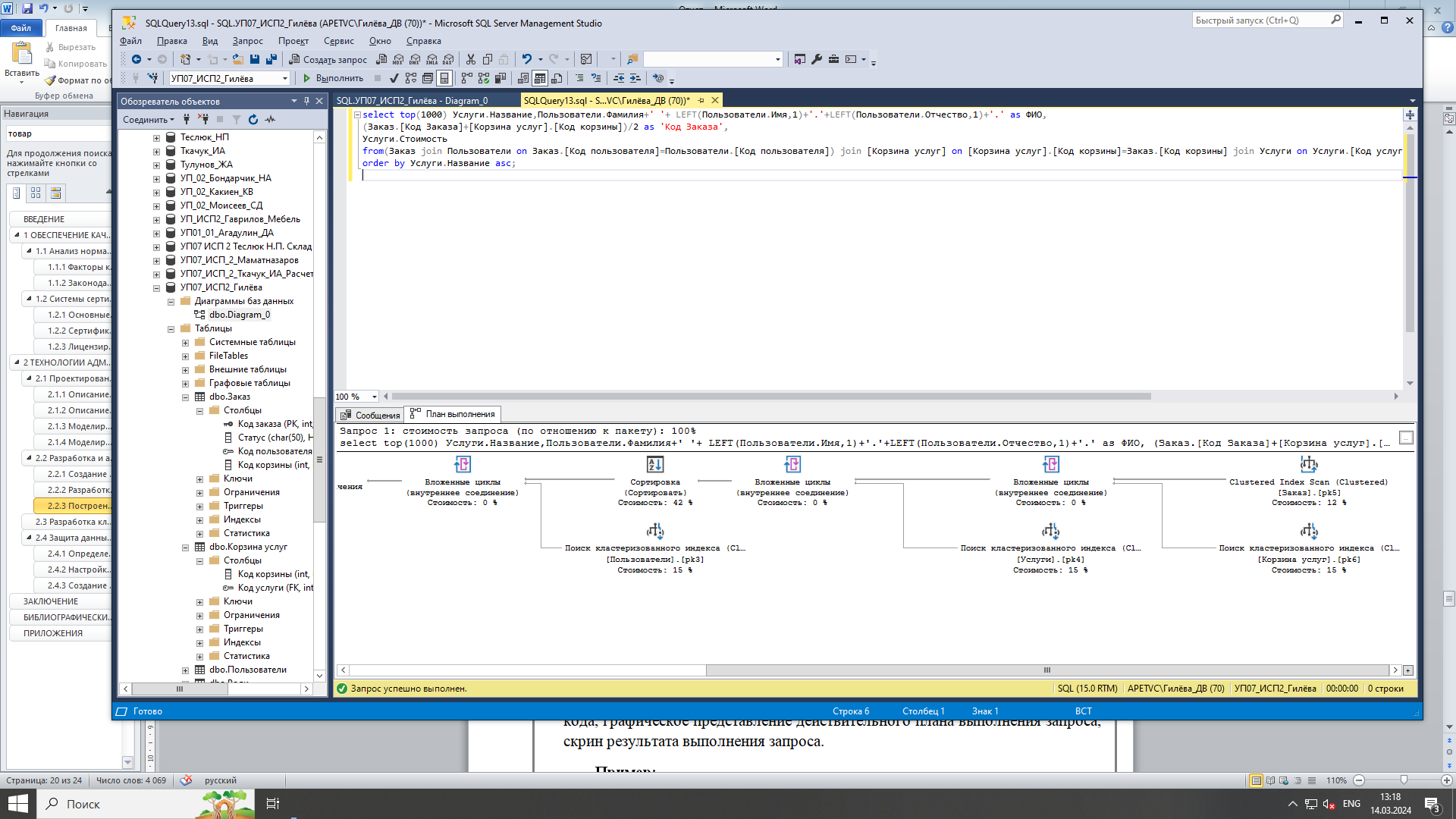
(Заказ.[Код Заказа]+[Корзина услуг].[Код корзины])/2 as [Код заказа],

Услуги.Стоимость

from(Заказ join Пользователи on Заказ.[Код пользователя]=Пользователи.[Код пользователя]) join [Корзина услуг] on [Корзина услуг].[Код корзины]=Заказ.[Код корзины] join Услуги on Услуги.[Код услуги]=[Корзина услуг].[Код корзины]

order by Услуги.Название asc;



 Рисунок 4 – План выполнения запроса

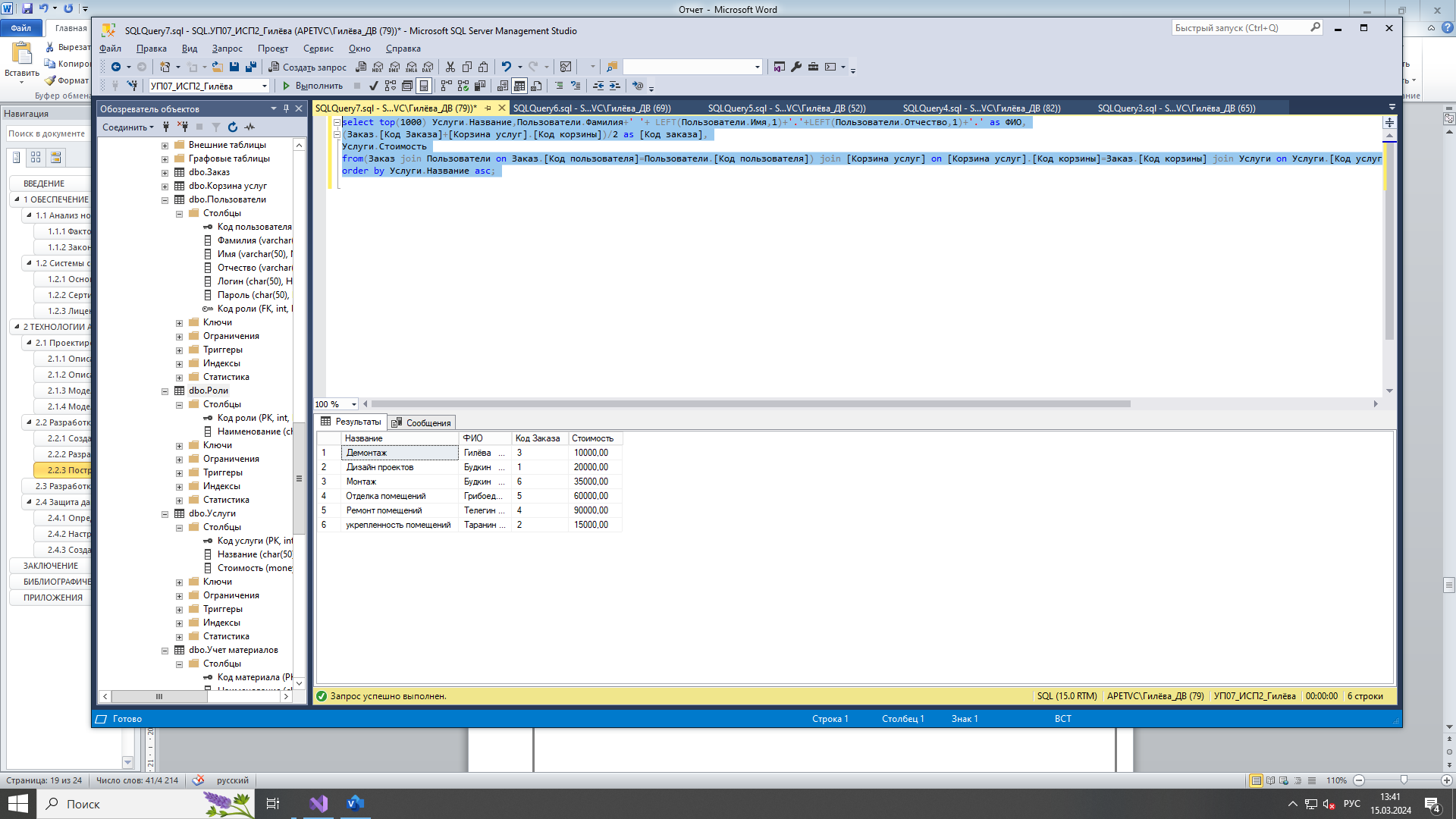


Рисунок 5 – Результат выполнения запроса

1. Вывести список клиентов, приобретённых ими услуг, стоимостью более 40000 руб. Предполагаемый план выполнения представлен на рисунке 6, результат выполнения – на рисунке 7.

CREATE VIEW запрос2 as

select top(1000) Услуги.Название,Пользователи.Фамилия+' '+ LEFT(Пользователи.Имя,1)+'.'+LEFT(Пользователи.Отчество,1)+'.' as ФИО,

(Заказ.[Код Заказа]+[Корзина услуг].[Код корзины])/2 as [Код заказа],

Услуги.Стоимость

from(Заказ join Пользователи on Заказ.[Код пользователя]=Пользователи.[Код пользователя]) join [Корзина услуг] on [Корзина услуг].[Код корзины]=Заказ.[Код корзины] join Услуги on Услуги.[Код услуги]=[Корзина услуг].[Код корзины]

where Услуги.Стоимость>40000

order by Услуги.Название asc;

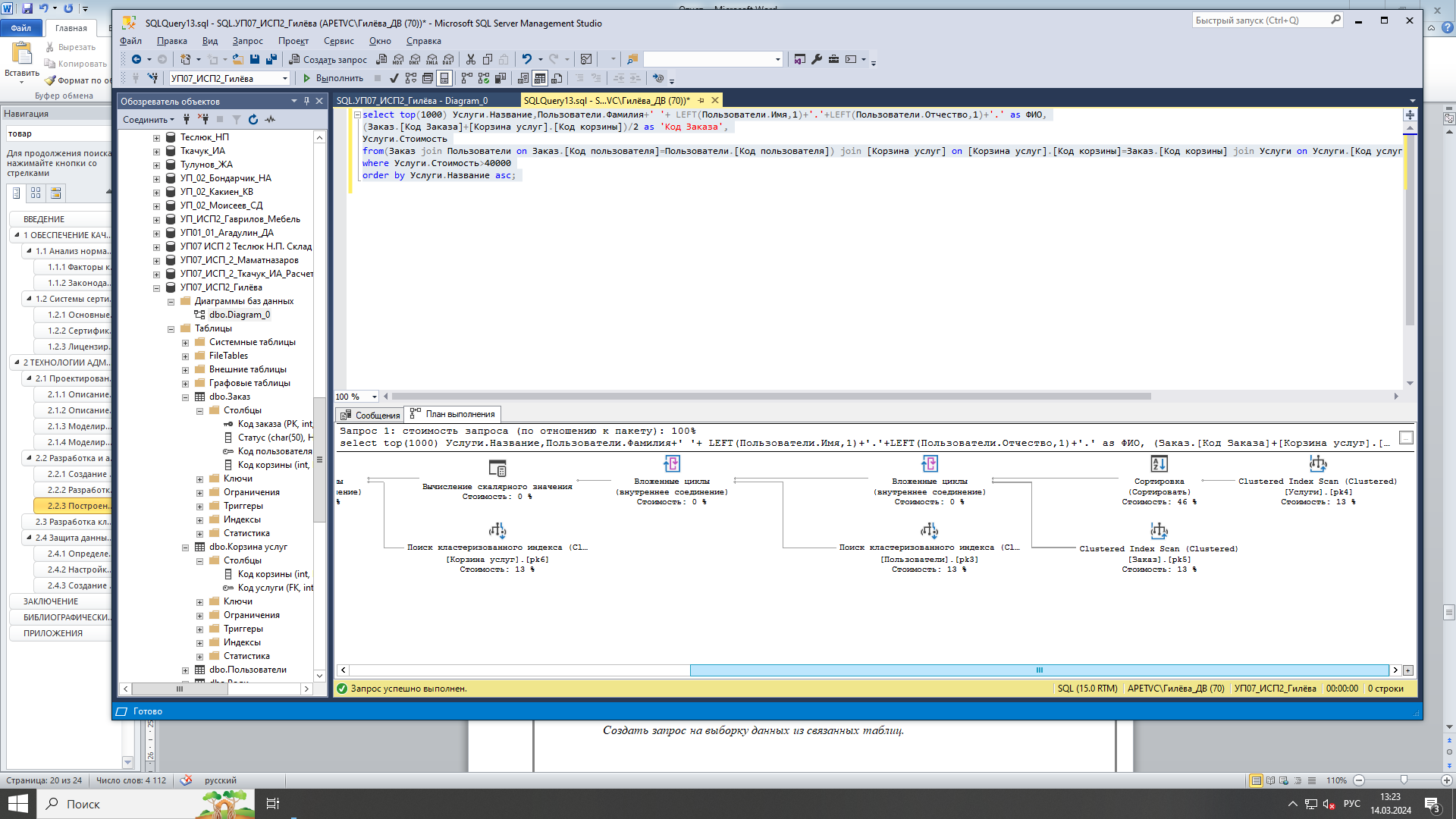
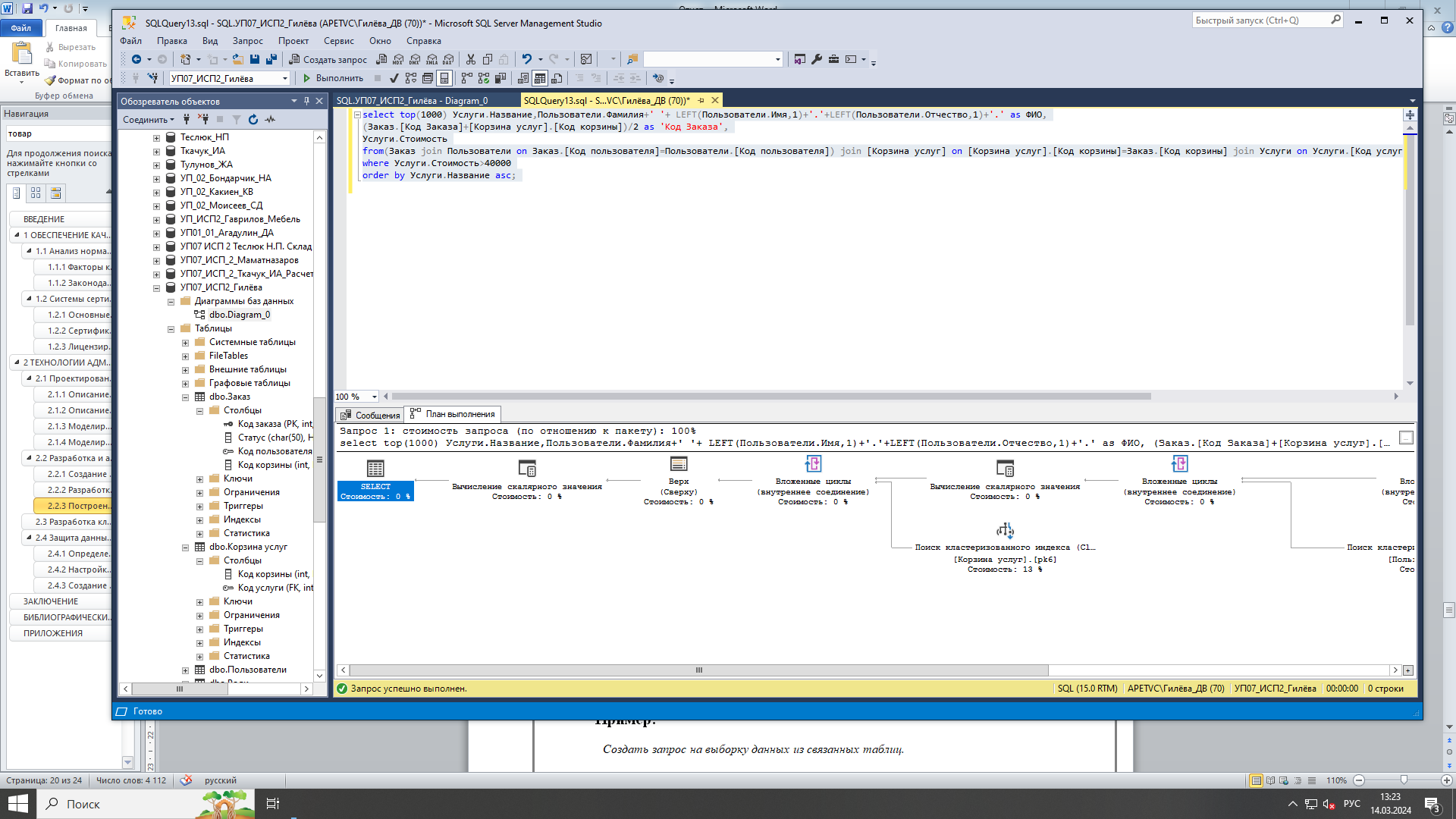


Рисунок 6 – План выполнения запроса

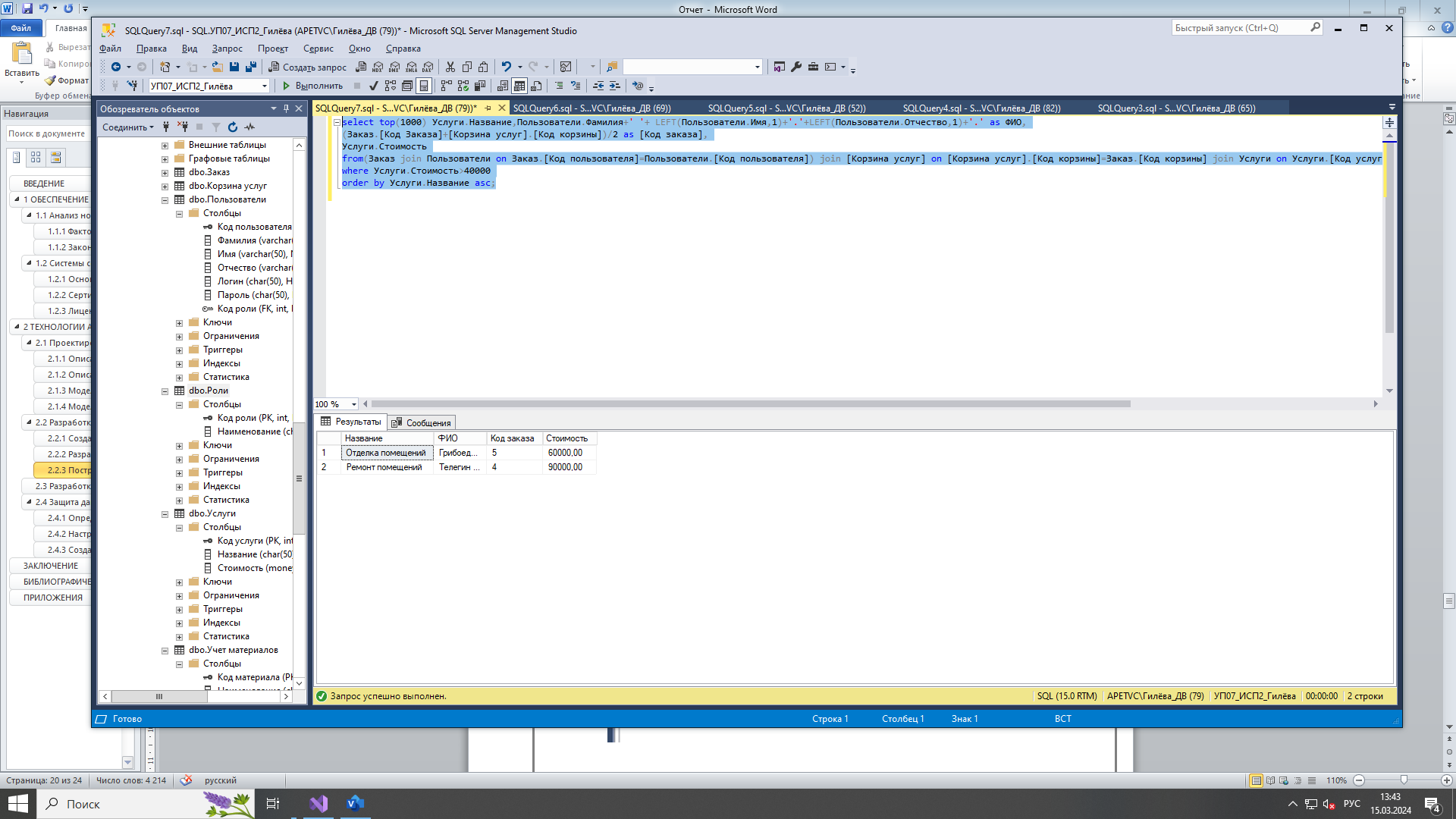


Рисунок 7 – Результат выполнения запроса

1. Провести подсчет всех видов материалов и клиентов их купивших. С группировкой по нескольким полям. Предполагаемый план выполнения представлен на рисунке 8, результат выполнения – на рисунке 9.

select top(1000)

[Учет материалов].Наименование, [Учет материалов].Цена,[Учет материалов].[Кол-во], Пользователи.Фамилия+' '+ LEFT(Пользователи.Имя,1)+'.'+LEFT(Пользователи.Отчество,1)+'.' as ФИО, (Заказ.[Код заказа]+[Корзина услуг].[Код корзины])/2 as [код заказа], Услуги.[Название]

from [Учет материалов] join Услуги on Услуги.[Код услуги]=[Учет материалов].[Код услуги] join [Корзина услуг] on [Корзина услуг].[Код услуги]=Услуги.[Код услуги] join Заказ on [Корзина услуг].[Код корзины]=Заказ.[Код корзины] join Пользователи on Пользователи.[Код пользователя]=Заказ.[Код пользователя]

order by [Учет материалов].Наименование

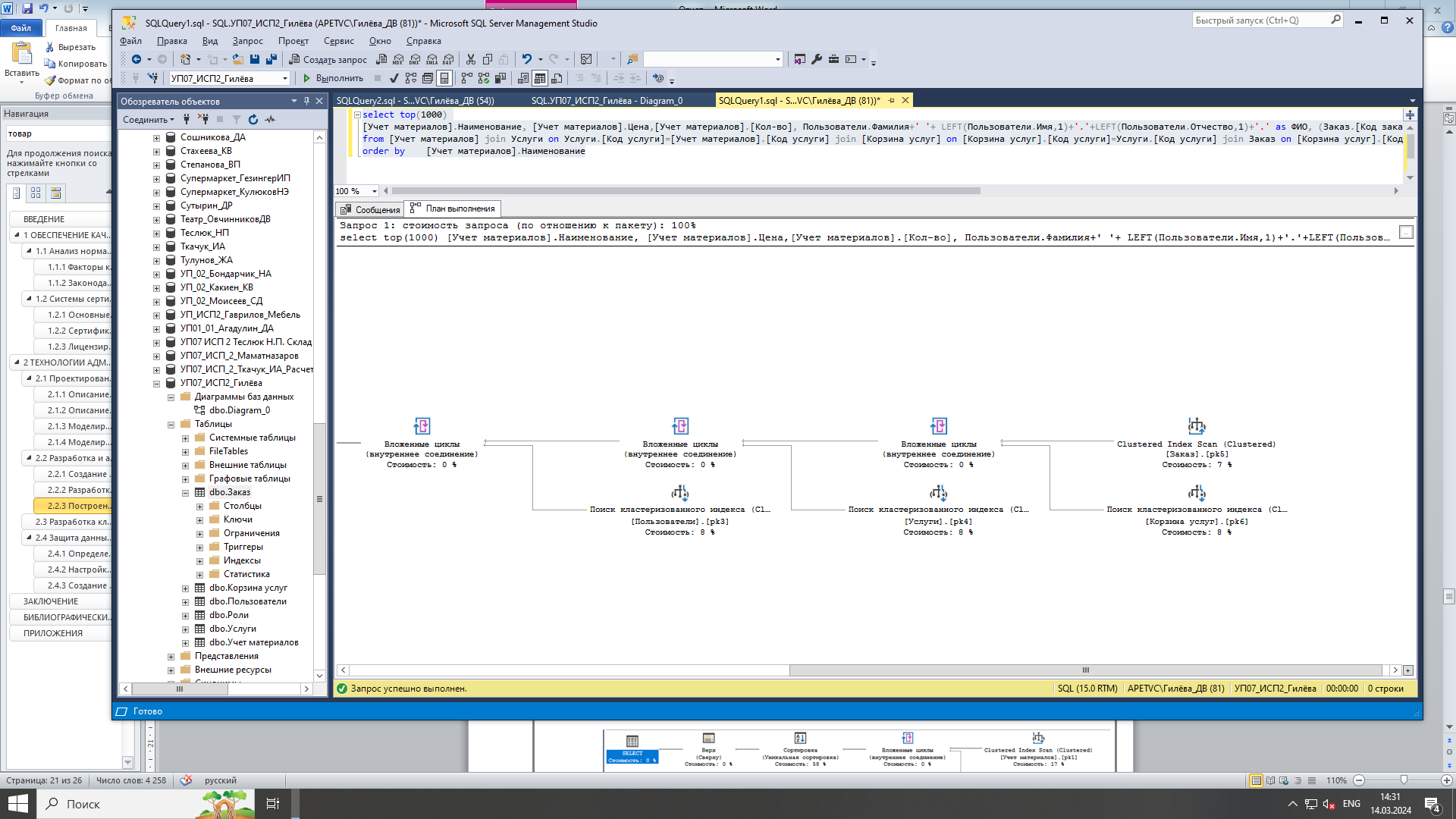
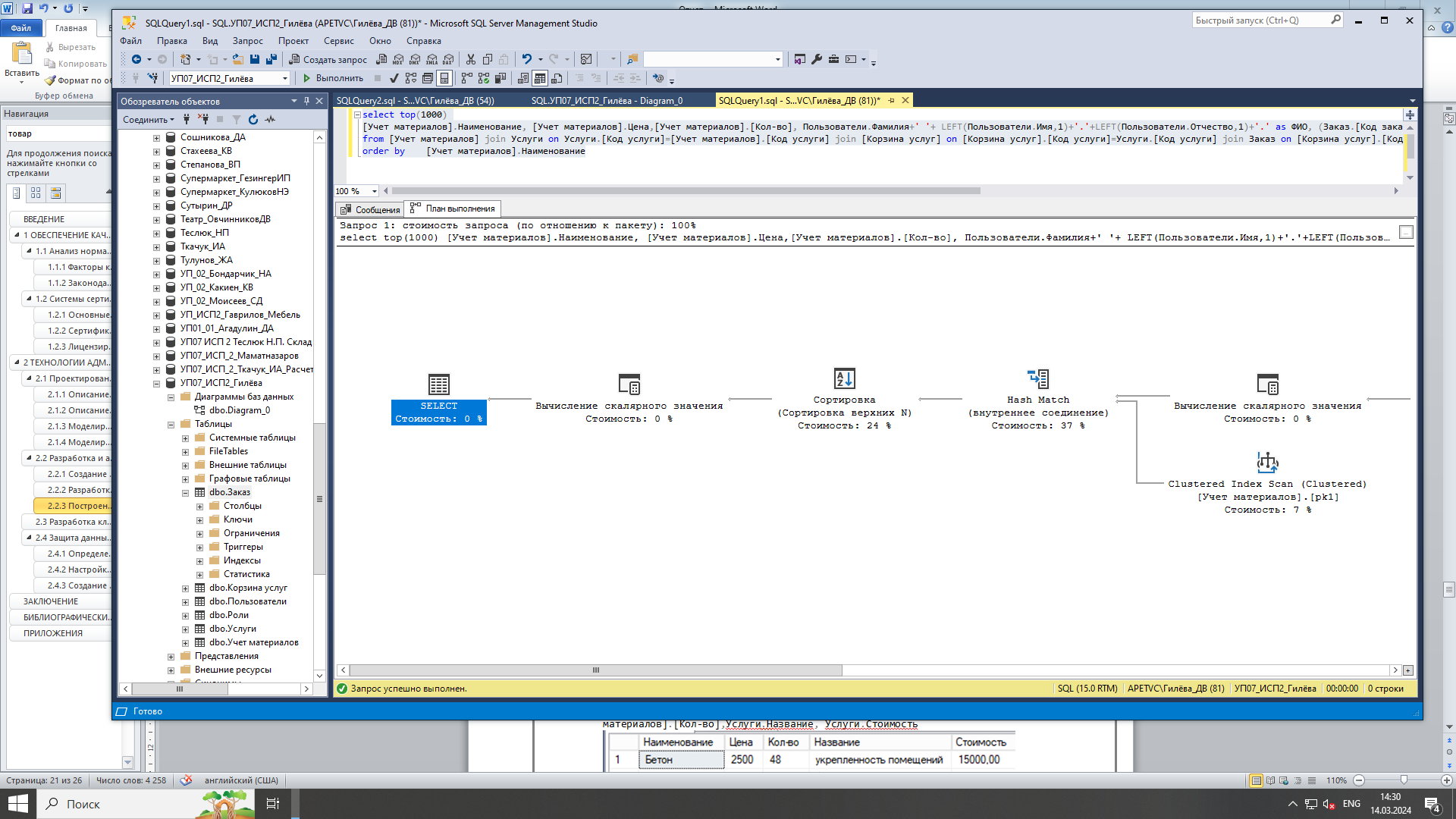


Рисунок 8– План выполнения запроса

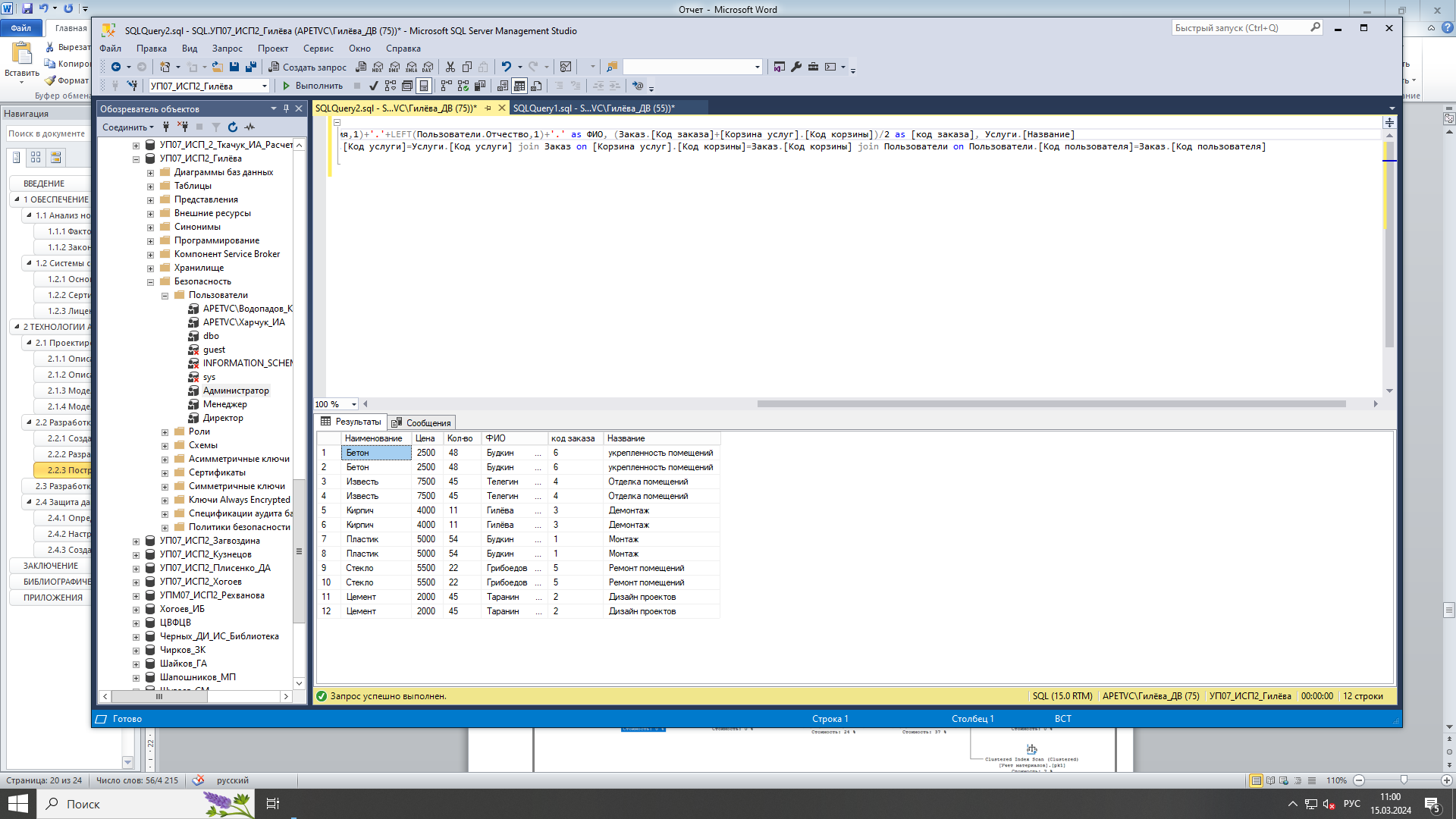


Рисунок 9 – Результат выполнения запроса

1. Вывести список материалов для каждой услуги и их среднюю стоимость, используя группировку данных по нескольким полям. Предполагаемый план выполнения представлен на рисунке 10, результат выполнения – на рисунке 11.

create view Запрос4 as

select top(1000)

[Учет материалов].Наименование, [Учет материалов].Цена,[Учет материалов].[Кол-во],Услуги.Название, Услуги.Стоимость

from [Учет материалов] join Услуги on Услуги.[Код услуги]=[Учет материалов].[Код услуги]

group by [Учет материалов].Наименование, [Учет материалов].Цена,[Учет материалов].[Кол-во],Услуги.Название, Услуги.Стоимость

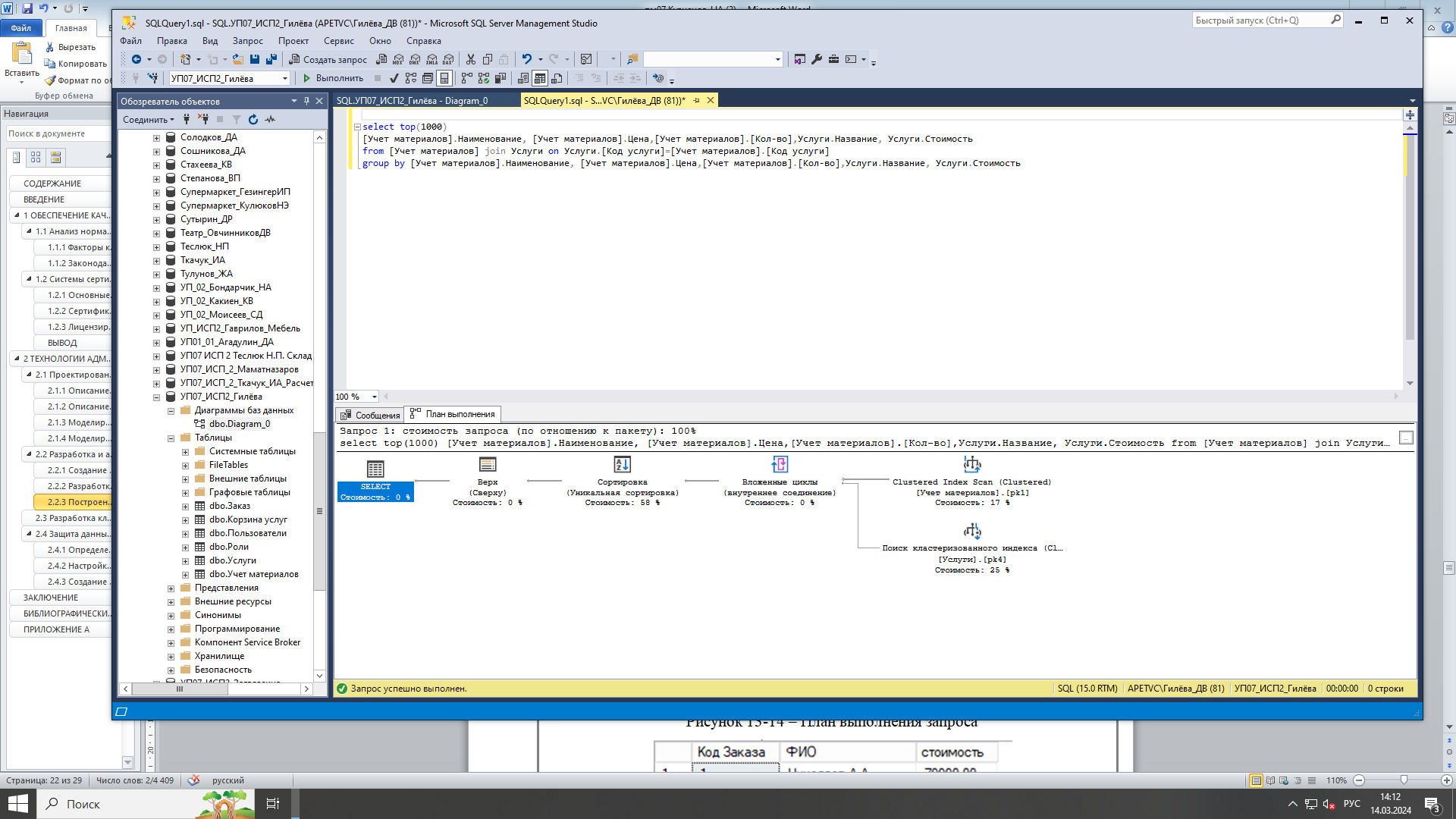


Рисунок 10 – План выполнения запроса

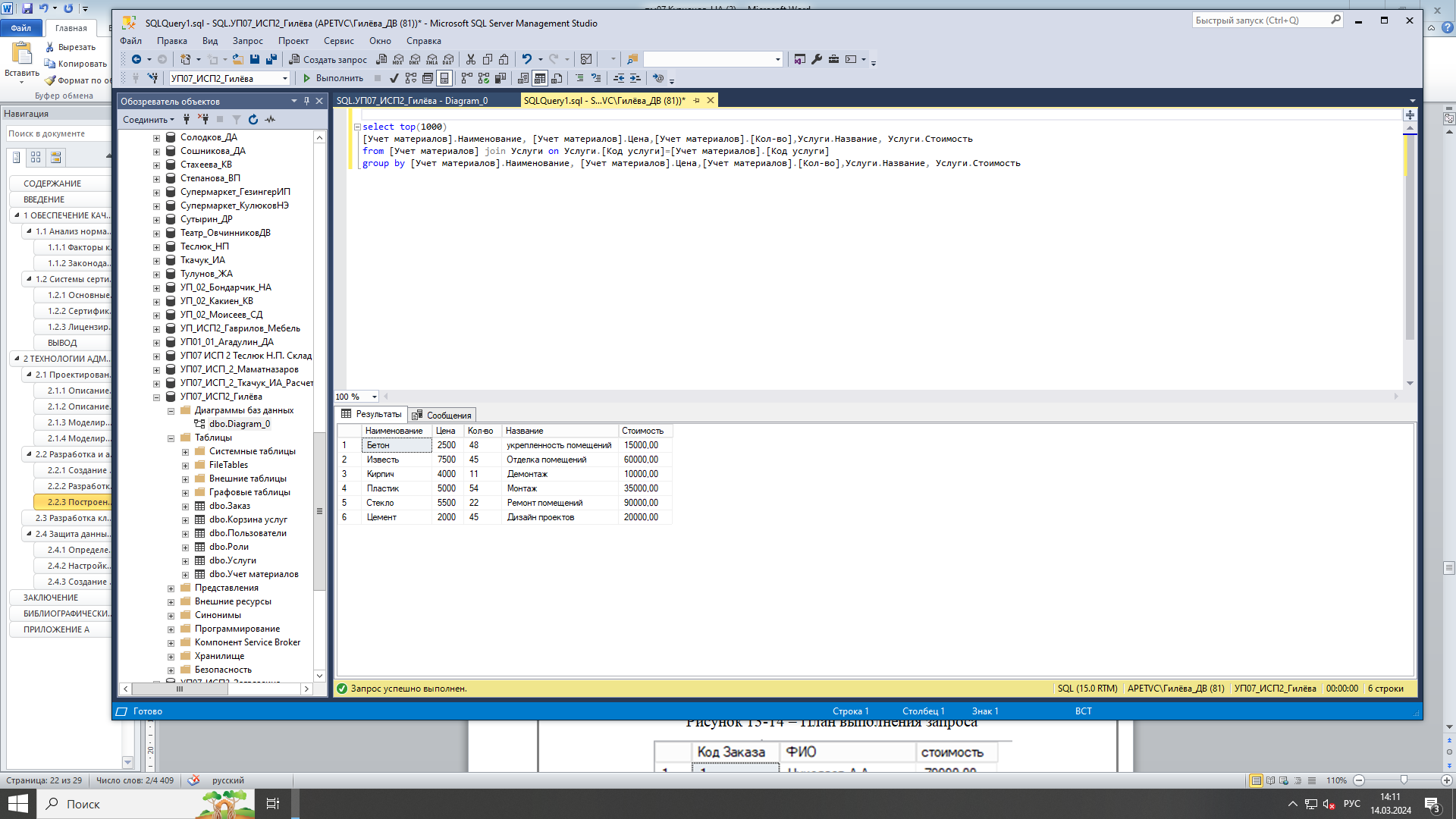


Рисунок 11 – Результат выполнения запроса

1. Вывести список услуг, средний чек и стоимость покупки с использованием агрегатных функций (SUM, AVG, MAX, MIN, COUNT). Предполагаемый план выполнения представлен на рисунке 12, результат выполнения – на рисунке 13.

CREATE VIEW Запрос5 as

select top(1000)

Услуги.Название,Пользователи.Фамилия+' '+ LEFT(Пользователи.Имя,1)+'.'+LEFT(Пользователи.Отчество,1)+'.' as ФИО,

AVG(Заказ.[Код Заказа]+[Корзина услуг].[Код корзины])/2 as 'средний чек',

Услуги.Стоимость

from(Заказ join Пользователи on Заказ.[Код пользователя]=Пользователи.[Код пользователя]) join [Корзина услуг] on [Корзина услуг].[Код корзины]=Заказ.[Код корзины] join Услуги on Услуги.[Код услуги]=[Корзина услуг].[Код корзины]

where Услуги.Стоимость>40000

group by Услуги.Название, Пользователи.Фамилия+' '+ LEFT(Пользователи.Имя,1)+'.'+LEFT(Пользователи.Отчество,1)+'.',Услуги.Стоимость

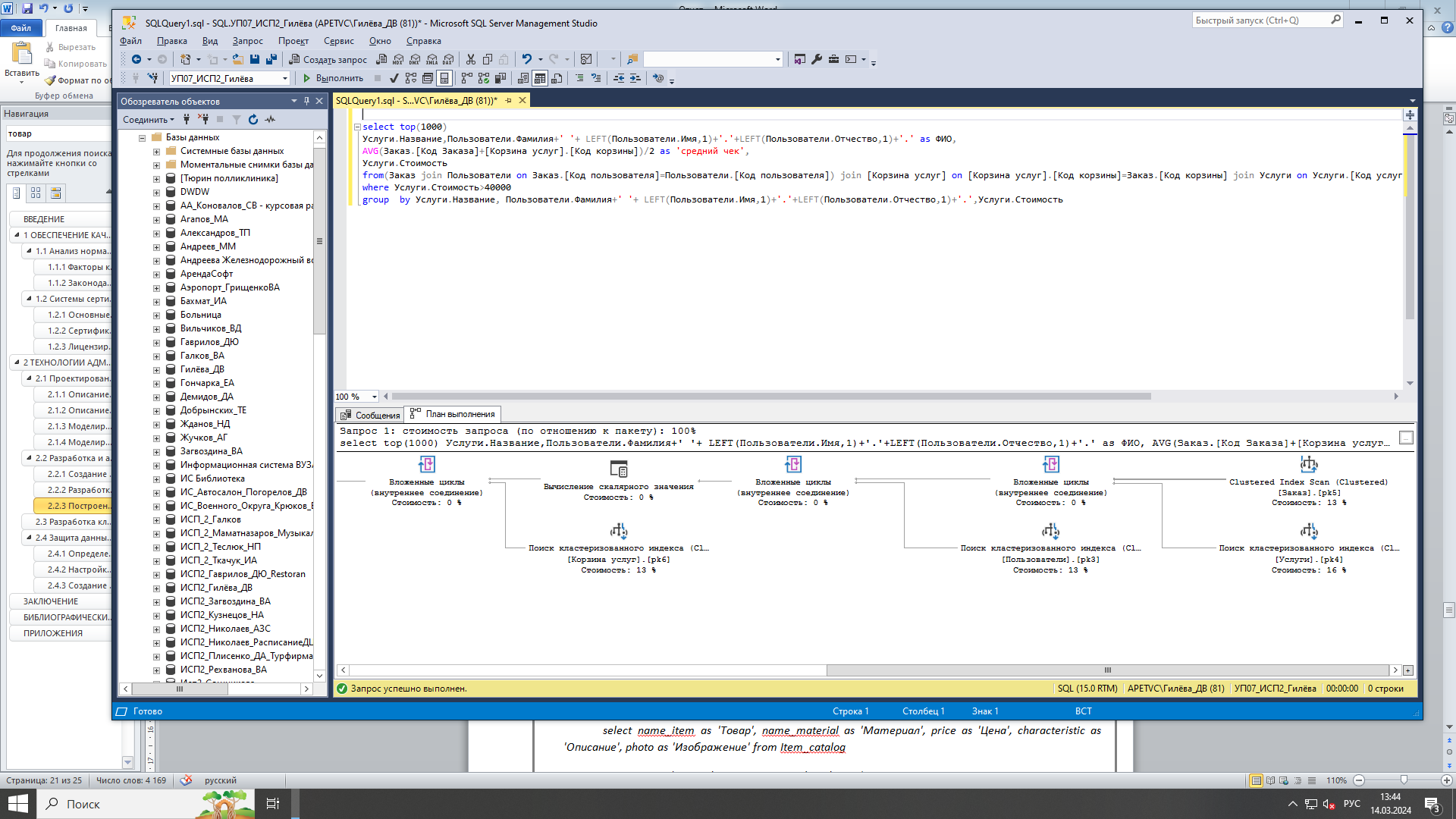
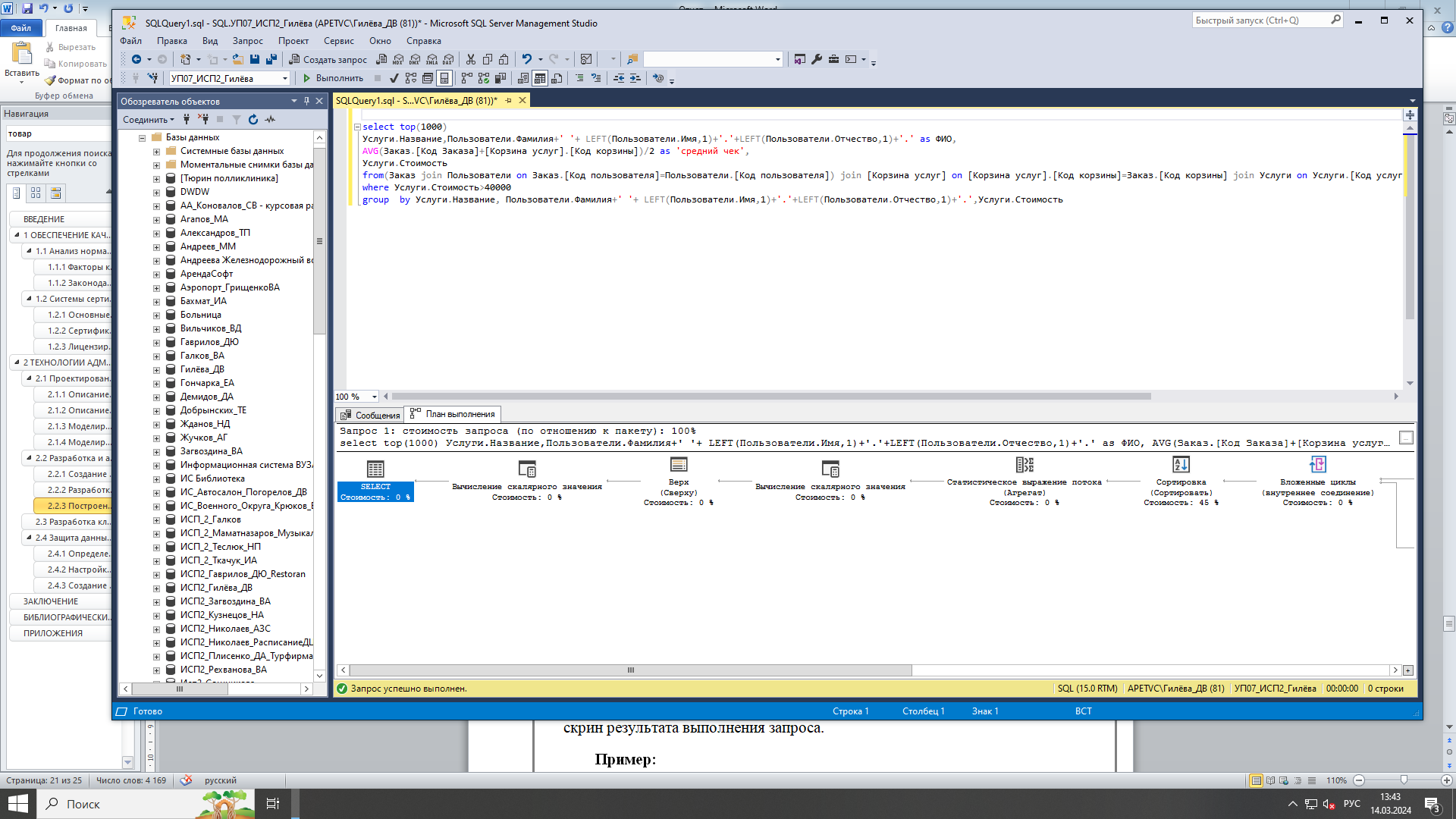


Рисунок 12– План выполнения запроса

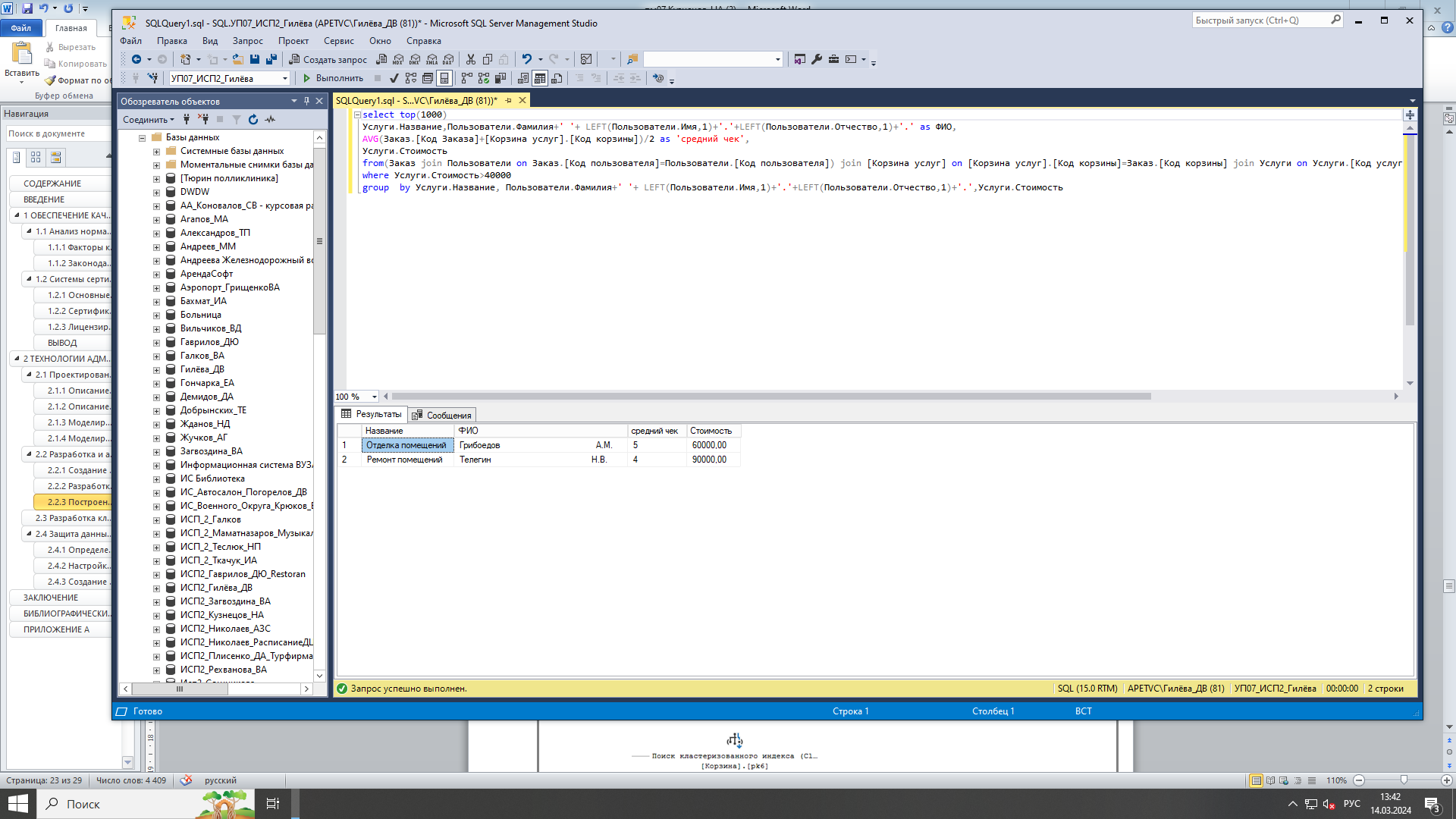


Рисунок 13 – Результат выполнения запроса

2.3 Разработка клиентской части приложения в инструментальной среде

Для реализации подключения базы данных используется Visual Studio C#.

В отчете описать процесс интеграции базы данных, в приложение, проиллюстрировав его рисунками.

Для интегрированной базы данных были созданы интерфейс администратора, директора, и интерфейс менеджера (форму авторизации, формы навигации по приложению и др.).

Все разработанные формы продемонстрированы на рисунках 14-16 в отчете, проиллюстрировав его рисунками и таблицами.

Таблица 2 – Свойства объектов формы авторизации

| Объекты | Свойства | Значения |
| --- | --- | --- |
| Form1 | name | Form1 |
| BackgroundImage | System.Drawing.Bitmap |
| FormBarderStyle | Sizable |
| Size | 372; 499 |
| TextBox1 | anchor | Top, Right |
| Font | Microsoft Sans Serif; 8,25pt |
| Visible | True |
| Size | 158; 20 |
| maskedTextBox1 | anchor | Top, Right |
| Size | 158; 20 |
| Font | Microsoft Sans Serif; 8,25pt |
| Visible | True |
| pictureBox1 | BackGroundImage | System.Drawing.Bitmap |
| Size | 100; 50 |
| textBox2 | anchor | Top, Left |
| Size | 145; 20 |
| Font | Microsoft Sans Serif; 8,25pt |
| Visible | True |
| textBox3 | anchor | Top, Left |
| Size | 145; 20 |
| Font | Microsoft Sans Serif; 8,25pt |
| Visible | True |
| textBox4 | anchor | Top, Left |
| Size | 145; 20 |
| Font | Microsoft Sans Serif; 8,25pt |
| Visible | True |
| textBox5 | anchor | Top, Left |
| Size | 145; 20 |
| Font | Microsoft Sans Serif; 8,25pt |
| Visible | True |
| panel1 | anchor | Top, Left |
| Size | 273; 230 |
| Location | 76; 29 |
| button1 | anchor | Top, Right |
| Size | 93; 23 |
| Visible | True |
| button2 | anchor | Top, Right |
| Size | 93; 23 |
| Visible | True |
| Button3 | anchor | Top, Right |
| Size | 93; 23 |
| Visible | True |

Таблица 3 – Событийно-управляемые процедуры

| Объект | Событийно-управляемые процедуры | Код |
| --- | --- | --- |
| button1 | private void button1\_Click(object sender, EventArgs e) | {  try  {  db db = new db();  db.openconn();  DataTable dt = new DataTable();  SqlDataAdapter adapter = new SqlDataAdapter();  SqlCommand command = new SqlCommand($"SELECT [Код роли] FROM Пользователи WHERE Логин='{textBox1.Text}' AND Пароль='{maskedTextBox1.Text}'", db.getconn());  adapter.SelectCommand = command;  adapter.Fill(dt);  if (dt.Rows.Count > 0)  {  string role = null;  foreach (DataRow row in dt.Rows)  {  role = row["Код роли"].ToString();  if (role.Contains("1"))  {  Form2 form2 = new Form2();  form2.Show();  }  if (role.Contains("3"))  {  Form3 form3 = new Form3();  form3.Show();  }  if (role.Contains("2"))  {  Form3 form3 = new Form3();  form3.Show();  }  db.closeconn();  }  }  }  catch (Exception ex)  {  MessageBox.Show(ex.Message);  } |
| textBox1 | private void textBox1\_Click(object sender, EventArgs e) | {  textBox1.Text = null;  textBox1.ForeColor = Color.Black;  } |
| maskedTextBox1 | private void maskedTextBox1\_Click(object sender, EventArgs e) | {  maskedTextBox1.Text = null;  maskedTextBox1.ForeColor = Color.Black;  maskedTextBox1.UseSystemPasswordChar = true;  } |
| Form1 | private void Form1\_Load(object sender, EventArgs e) | {  textBox1.Text = "Логин";//подсказка  textBox1.ForeColor = Color.Gray;  maskedTextBox1.Text = "Пароль";//подсказка  maskedTextBox1.ForeColor = Color.Gray;  textBox2.Text = "ФИО";//подсказка  textBox2.ForeColor = Color.Gray;  textBox3.Text = "Логин";//подсказка  textBox3.ForeColor = Color.Gray;  textBox4.Text = "Пароль";//подсказка  textBox4.ForeColor = Color.Gray;  textBox5.Text = "Повтор пароля";//подсказка  textBox5.ForeColor = Color.Gray;  } |
| textBox2 | private void textBox2\_Enter(object sender, EventArgs e) | {  textBox2.Text = null;  textBox2.ForeColor = Color.Black;  } |
| textBox3 | private void textBox3\_Enter(object sender, EventArgs e) | {  textBox3.Text = null;  textBox3.ForeColor = Color.Black;  } |
| textBox4 | private void textBox4\_Enter(object sender, EventArgs e) | {  textBox4.Text = null;  textBox4.ForeColor = Color.Black;  } |
| textBox5 | private void textBox5\_Enter(object sender, EventArgs e) | {  textBox5.Text = null;  textBox5.ForeColor = Color.Black;  } |
| button2 | private void button2\_Click(object sender, EventArgs e) | {  panel1.Show();  } |

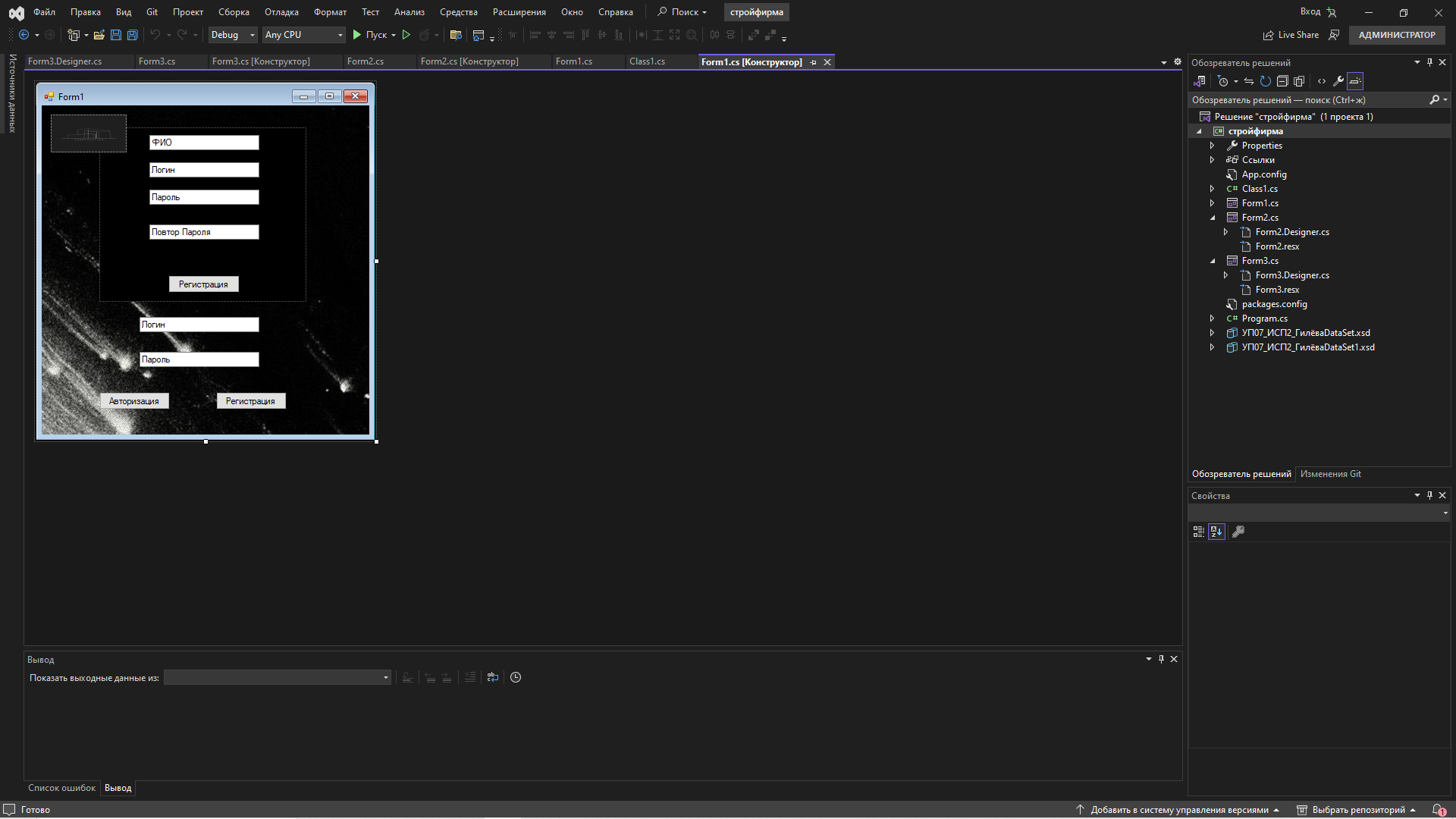


Рисунок 14 – Форма Пользователя (авторизации)

Таблица 4 – Свойства объектов формы директора

| Объекты | Свойства | Значения |
| --- | --- | --- |
| Form2 | name | Form2 |
| BackgroundImage | System.Drawing.Bitmap |
| FormBarderStyle | Sizable |
| Size | 816; 489 |
| dataGridView3 | anchor | Top, Left |
| AutoSizeColumnMode | None |
| DataSource | заказBindingSource |
| Size | 725; 370 |
| Visible | True |
| RowHeaderVisible | True |
| dataGridView1 | anchor | Top, Left |
| AutoSizeColumnMode | None |
| DataSource | пользователиBindingSource |
| Size | 725; 370 |
| Visible | True |
| RowHeaderVisible | True |
| dataGridView2 | anchor | Top, Left |
| AutoSizeColumnMode | None |
| DataSource | услугиBindingSource |
| Size | 725; 370 |
| Visible | True |
| RowHeaderVisible | True |

Таблица 5 – Событийно-управляемые процедуры

| Объект | Событийно-управляемые процедуры | Код |
| --- | --- | --- |
| Form2 | private void Form2\_Load(object sender, EventArgs e) | {  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП07\_ИСП2\_ГилёваDataSet1.Заказ". При необходимости она может быть перемещена или удалена.  this.заказTableAdapter.Fill(this.уП07\_ИСП2\_ГилёваDataSet1.Заказ);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП07\_ИСП2\_ГилёваDataSet1.Услуги". При необходимости она может быть перемещена или удалена.  this.услугиTableAdapter1.Fill(this.уП07\_ИСП2\_ГилёваDataSet1.Услуги);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП07\_ИСП2\_ГилёваDataSet.Услуги". При необходимости она может быть перемещена или удалена.  this.услугиTableAdapter.Fill(this.уП07\_ИСП2\_ГилёваDataSet.Услуги);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП07\_ИСП2\_ГилёваDataSet.Пользователи". При необходимости она может быть перемещена или удалена.  this.пользователиTableAdapter.Fill(this.уП07\_ИСП2\_ГилёваDataSet.Пользователи);  } |
| toolStripButton2 | private void toolStripButton2\_Click(object sender, EventArgs e) | {  dataGridView1.Visible = false;  dataGridView2.Visible = true;  } |
| toolStripButton3 | private void toolStripButton3\_Click(object sender, EventArgs e) | {  dataGridView3.Visible = true;  dataGridView1.Visible = false;  dataGridView2.Visible = false;  } |

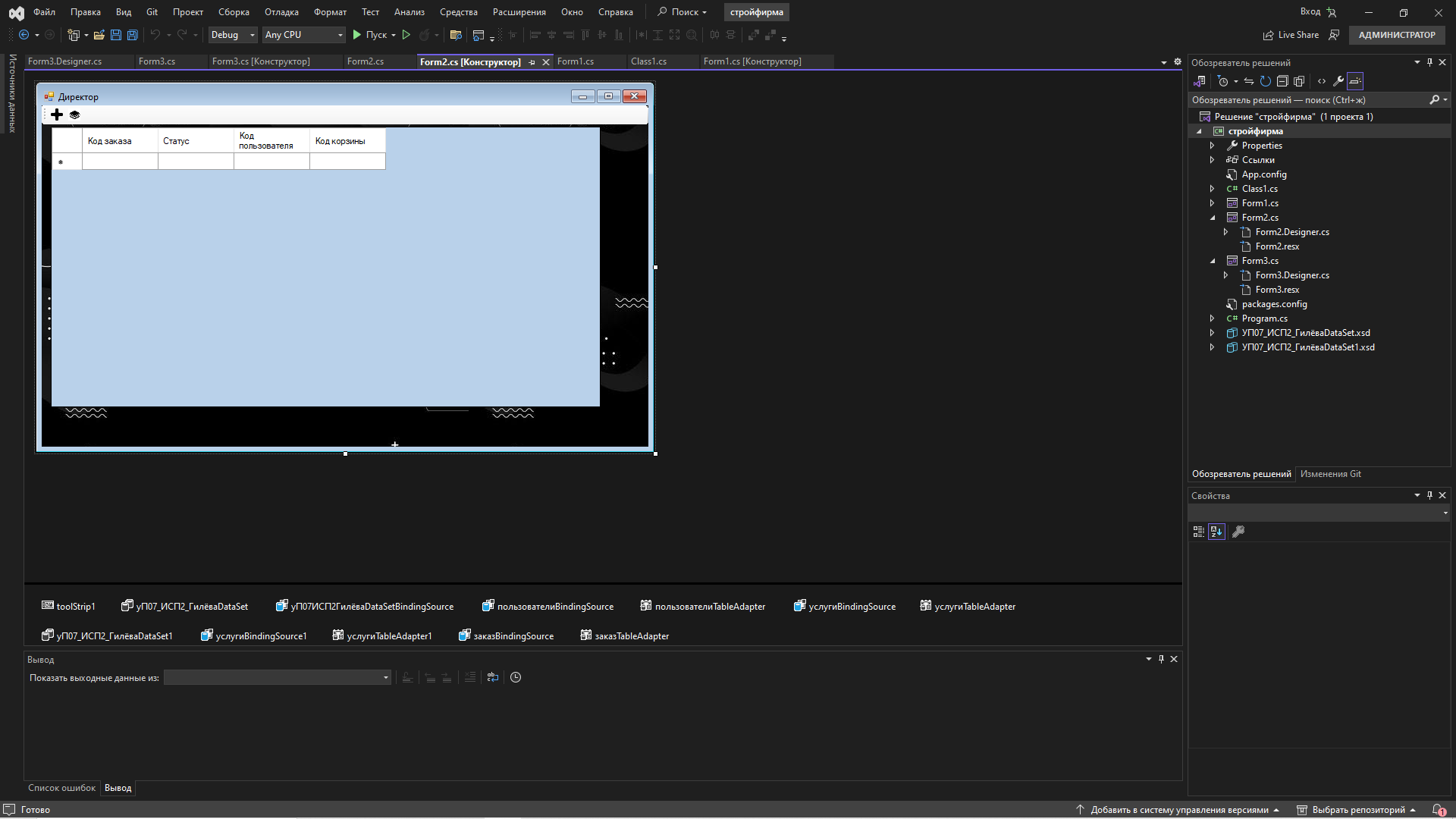


Рисунок 15 – Форма директора

Таблица 6 - Свойства объектов формы для менеджера и администратора

| Объекты | Свойства | Значения |
| --- | --- | --- |
| Form3 | name | Form3 |
| BackgroundImage | System.Drawing.Bitmap |
| FormBarderStyle | Sizable |
| Size | 949; 600 |
| dataGridView3 | anchor | Top, Left |
| AutoSizeColumnMode | None |
| DataSource | учётМатериаловBindingSource |
| Size | 638; 150 |
| Visible | True |
| RowHeaderVisible | True |
| dataGridView1 | anchor | Top, Left |
| AutoSizeColumnMode | DisplayedCells |
| DataSource | услугиBindingSource |
| Size | 613; 203 |
| Visible | True |
| RowHeaderVisible | True |
| dataGridView2 | anchor | Top, Left |
| AutoSizeColumnMode | None |
| DataSource | пользователиBindingSource |
| Size | 792; 265 |
| Visible | True |
| RowHeaderVisible | True |
| Panel1 | anchor | Top, Left |
| Size | 798; 317 |
| Location | 12; 237 |
| Panel2 | anchor | Top, Left |
|  | Size | 644; 290 |
| Location | 650; 159 |
| Panel3 | anchor | Top, Left |
| Size | 182; 184 |
| Location | 329; 38 |
| Panel4 | anchor | Top, Left |
| Size | 540; 158 |
| Location | 220; 139 |
| button1 | anchor | Top, Left |
| Size | 75; 23 |
| Visible | True |
| button2 | anchor | Top, Left |
| Size | 75; 23 |
| Visible | True |
| Button3 | anchor | Top, Left |
| Size | 75; 23 |
| Visible | True |
| Button4 | anchor | Top, Left |
| Size | 75; 23 |
| Visible | True |
| Button5 | anchor | Top, Left |
| Size | 75; 23 |
| Visible | True |
| Button6 | anchor | Top, Left |
| Size | 75; 23 |
| Visible | True |
| Button7 | anchor | Top, Left |
| Size | 75; 23 |
| Visible | True |
| Button8 | anchor | Top, Left |
| Size | 75; 23 |
| Visible | True |
| pictureBox1 | BackGroundImage | System.Drawing.Bitmap |
|  | Size | 23; 25 |
| pictureBox2 | BackGroundImage | System.Drawing.Bitmap |
| Size | 23; 25 |
| pictureBox3 | BackGroundImage | System.Drawing.Bitmap |
| Size | 23; 25 |
| pictureBox4 | BackGroundImage | System.Drawing.Bitmap |
| Size | 23; 25 |
| textBox1 | anchor | Top, Left |
| Location | 100; 20 |
| Visible | True |
| textBox2 | anchor | Top, Left |
| Location | 100; 20 |
| Visible | True |
| textBox3 | anchor | Top, Left |
| Location | 100; 20 |
| Visible | True |
| textBox4 | anchor | Top, Left |
| Location | 100; 20 |
| Visible | True |
| textBox5 | anchor | Top, Left |
| Location | 100; 20 |
| Visible | True |
| textBox6 | anchor | Top, Left |
| Location | 100; 20 |
| Visible | True |
| textBox7 | anchor | Top, Left |
| Location | 100; 20 |
| Visible | True |
| textBox8 | anchor | Top, Left |
| Location | 100; 20 |
| Visible | True |
| textBox9 | anchor | Top, Left |
| Location | 100; 20 |
|  | Visible | True |
| textBox10 | anchor | Top, Left |
| Location | 100; 20 |
| Visible | True |
| textBox11 | anchor | Top, Left |
| Location | 100; 20 |
| Visible | True |
| textBox12 | anchor | Top, Left |
| Location | 100; 20 |
| Visible | True |
| button1 | anchor | Top, Left |
| Size | 115; 25 |
| Visible | True |
| label1 | Location | 46; 26 |
| Size | 57; 13 |
| Visible | True |
| Label2 | Location | 46;72 |
| Size | 62; 13 |
| Visible | True |
| Label3 | Location | 18; 247 |
| Size | 33; 13 |
| Visible | True |
| Label4 | Location | 18; 21 |
| Size | 83; 13 |
| Visible | True |
| Label5 | Location | 180; 247 |
| Size | 62; 13 |
| Visible | True |
| Label6 | Location | 180; 201 |
| Size | 66; 13 |
| Visible | True |
| Label7 | Location | 4; 13 |
| Size | 56; 13 |
| Visible | True |
| Label8 | Location | 4; 40 |
| Size | 29; 13 |
| Visible | True |
| Label9 | Location | 3; 73 |
| Size | 54; 13 |
| Visible | True |
| Label10 | Location | 184; 17 |
| Size | 38; 13 |
| Visible | True |
| label11 | Location | 184; 73 |
| Size | 54; 13 |
| Visible | True |
| label12 | Location | 184; 43 |
| Size | 45; 13 |
| Visible | True |

Таблица 7 – Событийно-управляемые процедуры

| Объект | Событийно-управляемые процедуры | Код |
| --- | --- | --- |
| Form3 | private void Form3\_Load(object sender, EventArgs e) | {  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП07\_ИСП2\_ГилёваDataSet1.Учет\_материалов". При необходимости она может быть перемещена или удалена.  this.учет\_материаловTableAdapter.Fill(this.уП07\_ИСП2\_ГилёваDataSet1.Учет\_материалов);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП07\_ИСП2\_ГилёваDataSet1.Пользователи". При необходимости она может быть перемещена или удалена.  this.пользователиTableAdapter.Fill(this.уП07\_ИСП2\_ГилёваDataSet1.Пользователи);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП07\_ИСП2\_ГилёваDataSet1.Услуги". При необходимости она может быть перемещена или удалена.  this.услугиTableAdapter.Fill(this.уП07\_ИСП2\_ГилёваDataSet1.Услуги);  } |
| pictureBox1 | private void pictureBox1\_Click(object sender, EventArgs e) | {  panel1.Hide();  } |
| toolStripButton4 | private void toolStripButton4\_Click(object sender, EventArgs e) | {  panel1.Show();  } |
| toolStripButton2 | private void toolStripButton2\_Click(object sender, EventArgs e) | {  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП07\_ИСП2\_ГилёваDataSet1.Пользователи". При необходимости она может быть перемещена или удалена.  this.пользователиTableAdapter.Fill(this.уП07\_ИСП2\_ГилёваDataSet1.Пользователи);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП07\_ИСП2\_ГилёваDataSet1.Услуги". При необходимости она может быть перемещена или удалена.  this.услугиTableAdapter.Fill(this.уП07\_ИСП2\_ГилёваDataSet1.Услуги);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП07\_ИСП2\_ГилёваDataSet1.Учет\_материалов". При необходимости она может быть перемещена или удалена.  this.учет\_материаловTableAdapter.Fill(this.уП07\_ИСП2\_ГилёваDataSet1.Учет\_материалов);  } |
| toolStripButton3 | private void toolStripButton3\_Click(object sender, EventArgs e) | {  panel3.Show();  } |
| pictureBox2 | private void pictureBox2\_Click(object sender, EventArgs e) | {  panel2.Hide();  } |
| toolStripButton5 | private void toolStripButton5\_Click(object sender, EventArgs e) | {  panel2.Show();  }  bool a = true;  int x, y; |
| pictureBox3 | private void pictureBox3\_Click(object sender, EventArgs e) | {  panel3.Hide();  } |
| panel2 | private void panel2\_MouseUp(object sender, MouseEventArgs e) | {  a = true;  } |
| panel2 | private void panel2\_MouseMove(object sender, MouseEventArgs e) | {  if (!a)  {  Panel mPanel = (Panel)sender;  mPanel.Left += e.X - x;  mPanel.Top += e.Y - y;  }  } |
| panel2 | private void panel2\_MouseDown(object sender, MouseEventArgs e) | {  Panel mPanel = (Panel)sender;  // стартовая позиция  x = e.X;  y = e.Y;  a = false;//переменная , нажата ли кнопка мыши  } |
| panel3 | private void panel3\_MouseUp(object sender, MouseEventArgs e) | {  a = true;  } |
| panel3 | private void panel3\_MouseMove(object sender, MouseEventArgs e) | {  if (!a)  {  Panel mPanel = (Panel)sender;  mPanel.Left += e.X - x;  mPanel.Top += e.Y - y;  }  } |
| panel3 | private void panel3\_MouseDown(object sender, MouseEventArgs e) | {  Panel mPanel = (Panel)sender;  // стартовая позиция  x = e.X;  y = e.Y;  a = false;//переменная , нажата ли кнопка мыши  } |
| button5 | private void button5\_Click(object sender, EventArgs e) | {  db db = new db();  db.openconn();  if (textBox1.Text == "" || textBox2.Text == "" )  {  MessageBox.Show("Заполните все поля!");  }  else  {  SqlCommand command = new SqlCommand("Insert into Услуги(Название, Стоимость) VALUES ( @a,@b)", db.getconn());  {  command.Parameters.AddWithValue("@a", textBox1.Text);  command.Parameters.AddWithValue("@b", textBox2.Text);  command.ExecuteNonQuery();  MessageBox.Show("Товар Добавлен");  db.closeconn();  this.услугиTableAdapter.Fill(this.уП07\_ИСП2\_ГилёваDataSet1.Услуги);  }  }  } |
| button7 | private void button7\_Click(object sender, EventArgs e) | {  db db = new db();  db.openconn();  if (textBox4.Text == "" || textBox3.Text == "" || textBox6.Text == "" || textBox5.Text == "" )  {  MessageBox.Show("Заполните все поля!");  }  else  {  SqlCommand command = new SqlCommand("Insert into [Учет материалов](Наименование, Цена, [Кол-во], [Код услуги]) VALUES ( @a,@b, @c, @d)", db.getconn());  {  command.Parameters.AddWithValue("@a", textBox4.Text);  command.Parameters.AddWithValue("@b", textBox3.Text);  command.Parameters.AddWithValue("@c", textBox6.Text);  command.Parameters.AddWithValue("@d", textBox5.Text);  command.ExecuteNonQuery();  MessageBox.Show("материал Добавлен");  db.closeconn();  this.учет\_материаловTableAdapter.Fill(this.уП07\_ИСП2\_ГилёваDataSet1.Учет\_материалов);  }  }  } |
| button4 | private void button4\_Click(object sender, EventArgs e) | {  panel3.Show();  } |
| button1 | private void button1\_Click(object sender, EventArgs e) | {  panel4.Show();  } |
| pictureBox4 | private void pictureBox4\_Click(object sender, EventArgs e) | {  panel4.Hide();  } |
| button8 | private void button8\_Click(object sender, EventArgs e) | {  db db = new db();  db.openconn();  if (textBox7.Text == "" || textBox8.Text == "" || textBox9.Text == "" || textBox10.Text == "" || textBox11.Text == "" || textBox12.Text == "")  {  MessageBox.Show("Заполните все поля!");  }  else  {  SqlCommand command = new SqlCommand("Insert into Пользователи (Фамилия,Имя,Отчество, Логин, Пароль, [Код Роли]) VALUES ( @a, @b, @c, @d, @e, @f)", db.getconn());  {  command.Parameters.AddWithValue("@a", textBox7.Text);  command.Parameters.AddWithValue("@b", textBox8.Text);  command.Parameters.AddWithValue("@c", textBox9.Text);  command.Parameters.AddWithValue("@d", textBox10.Text);  command.Parameters.AddWithValue("@e", textBox11.Text);  command.Parameters.AddWithValue("@f", textBox12.Text);  command.ExecuteNonQuery();  MessageBox.Show("Пользователь добавлен");  db.closeconn();  this.пользователиTableAdapter.Fill(this.уП07\_ИСП2\_ГилёваDataSet1.Пользователи);  }  }  } |
| button3 | private void button3\_Click(object sender, EventArgs e) | {  db db = new db();  db.openconn();  string query = ("DELETE From Услуги Where [Код услуги]=@id");  SqlCommand command = new SqlCommand(query, db.getconn());  command.Parameters.AddWithValue("@id", dataGridView1[0, dataGridView1.CurrentRow.Index].Value.ToString());  command.ExecuteNonQuery();  MessageBox.Show("Запись удалена");  db.closeconn();  this.услугиTableAdapter.Fill(this.уП07\_ИСП2\_ГилёваDataSet1.Услуги);  } |
| panel1 | private void panel1\_MouseUp(object sender, MouseEventArgs e) | {  a = true;  } |
| panel1 | private void panel1\_MouseMove(object sender, MouseEventArgs e) | {  if (!a)  {  Panel mPanel = (Panel)sender;  mPanel.Left += e.X - x;  mPanel.Top += e.Y - y;  }  } |
| panel1 | private void panel1\_MouseDown(object sender, MouseEventArgs e) | {  Panel mPanel = (Panel)sender;  // стартовая позиция  x = e.X;  y = e.Y;  a = false;//переменная , нажата ли кнопка мыши  } |
| panel4 | private void panel4\_MouseUp(object sender, MouseEventArgs e) | {  a = true;  } |
| panel4 | private void panel4\_MouseMove(object sender, MouseEventArgs e) | {  if (!a)  {  Panel mPanel = (Panel)sender;  mPanel.Left += e.X - x;  mPanel.Top += e.Y - y;  }  } |
| panel4 | private void panel4\_MouseDown(object sender, MouseEventArgs e) | {  Panel mPanel = (Panel)sender;  // стартовая позиция  x = e.X;  y = e.Y;  a = false;//переменная , нажата ли кнопка мыши  } |
| button6 | private void button6\_Click(object sender, EventArgs e) | {  db db = new db();  db.openconn();  string query = ("DELETE From [Учет материалов] Where [Код материала]=@id");  SqlCommand command = new SqlCommand(query, db.getconn());  command.Parameters.AddWithValue("@id", dataGridView3[0, dataGridView3.CurrentRow.Index].Value.ToString());  command.ExecuteNonQuery();  MessageBox.Show("Запись удалена");  db.closeconn();  this.учет\_материаловTableAdapter.Fill(this.уП07\_ИСП2\_ГилёваDataSet1.Учет\_материалов);  } |
| button2 | private void button2\_Click(object sender, EventArgs e) | {  db db = new db();  db.openconn();  string query = ("DELETE From Пользователи Where [Код пользователя]=@id");  SqlCommand command = new SqlCommand(query, db.getconn());  command.Parameters.AddWithValue("@id", dataGridView2[0, dataGridView2.CurrentRow.Index].Value.ToString());  command.ExecuteNonQuery();  MessageBox.Show("Запись удалена");  db.closeconn();  this.пользователиTableAdapter.Fill(this.уП07\_ИСП2\_ГилёваDataSet1.Пользователи);  } |

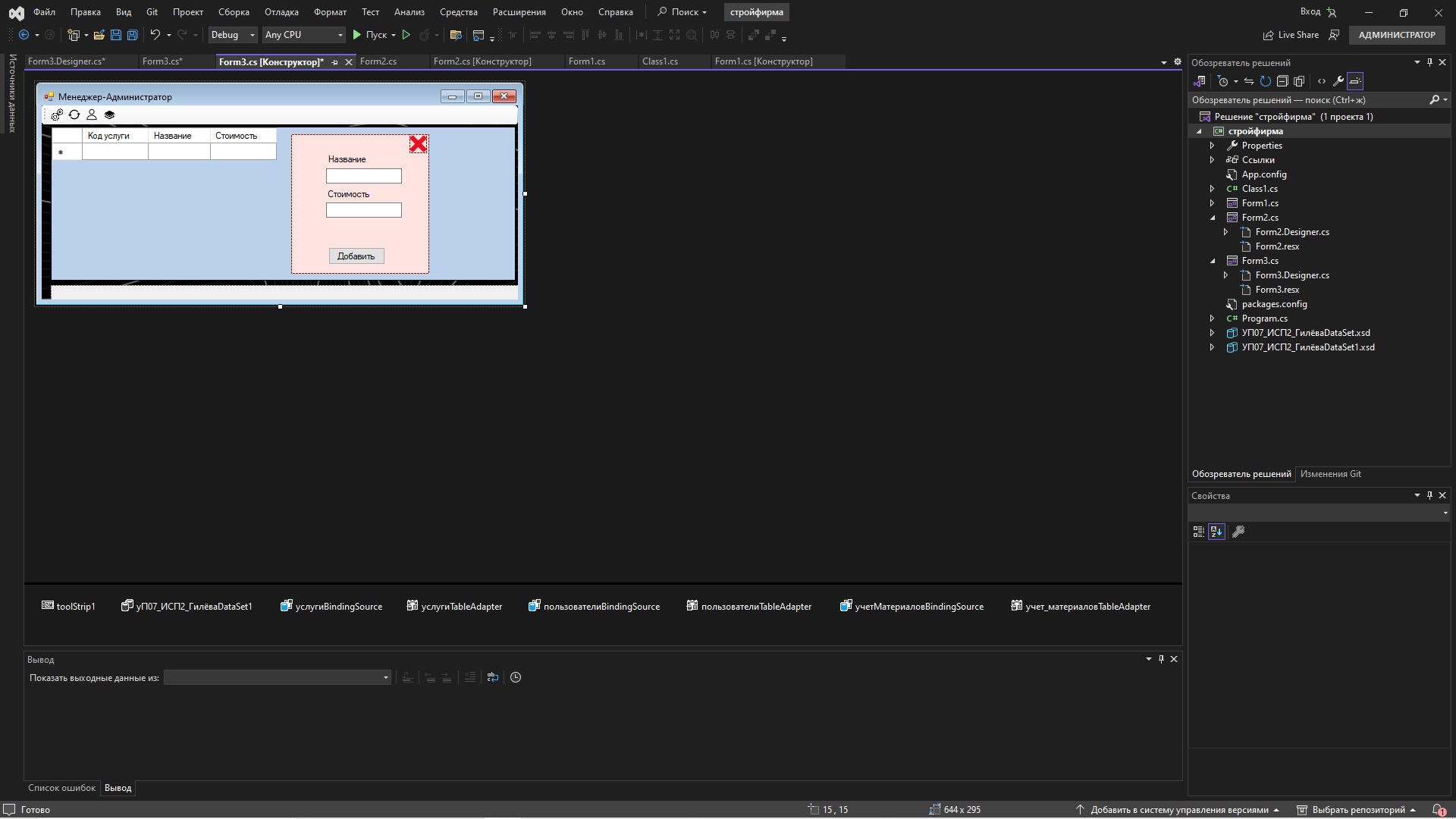


Рисунок 16 – Форма менеджера, администратора

2.4 Защита данных в хранилищах

2.4.1 Определение привилегий пользователей с различными ролями.

В SQL Server привилегии пользователей могут быть определены с помощью ролей, которые описывают набор разрешенных действий для определенных пользователей или групп пользователей. В SQL Server существуют такие роли как db\_datareader, db\_datawriter, db\_owner и другие, которые предоставляют различные уровни доступа к базе данных.

В соответствии с определенными в п.2.1.3, пользователями информационной системы, разграничены права доступа к приложению.

Созданы три имени входа на SQL Server: менеджер, директор, администратор. В созданной по индивидуальному заданию базе данных, созданы три одноименных пользователя.

Были созданы три пользователя для базы данных, разграничивающие права для каждого из зарегистрированных пользователей для клиентской части приложения. Процесс создания пользователей показан на рисунках 17-22.

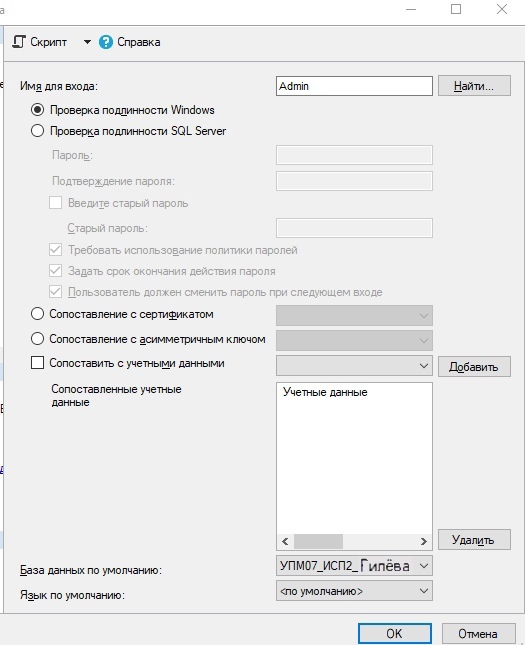


Рисунок 17 – Создание имени входа Администратора

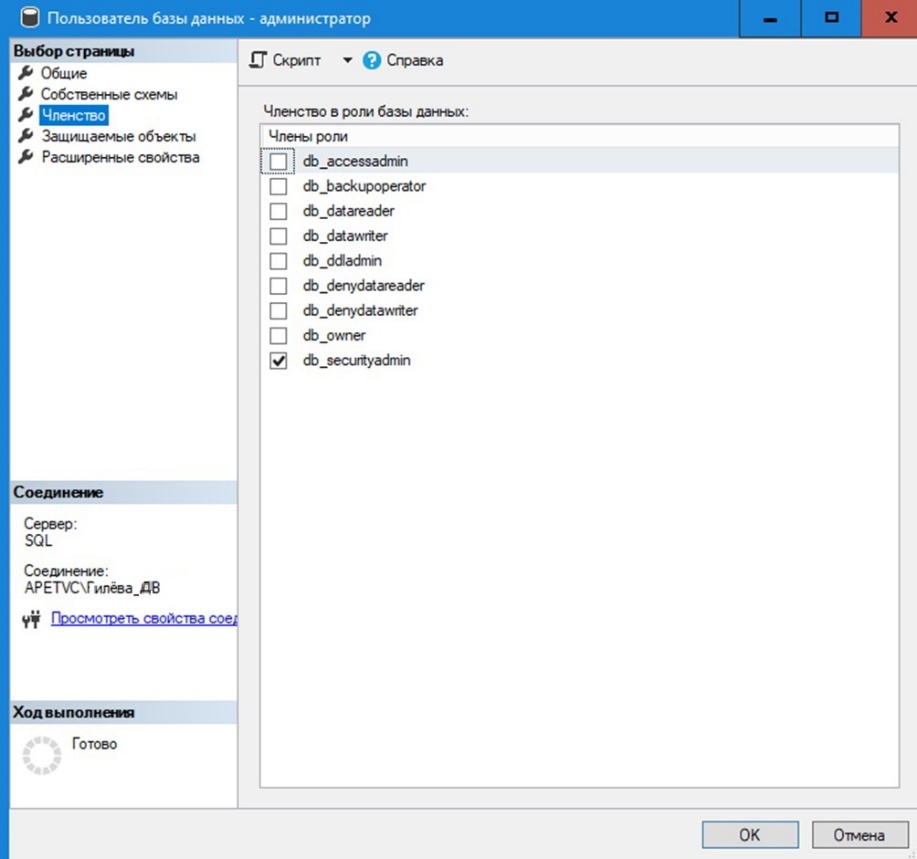


Рисунок 18 – Создание разграничения роли Администратор

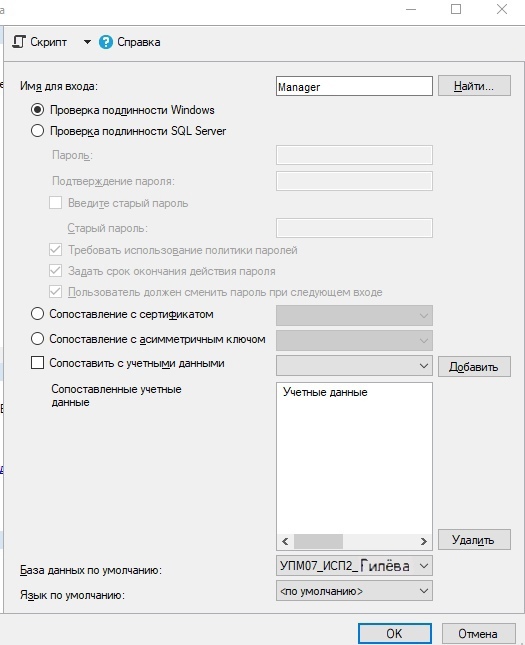


Рисунок 19 - Создание имени входа Менеджер

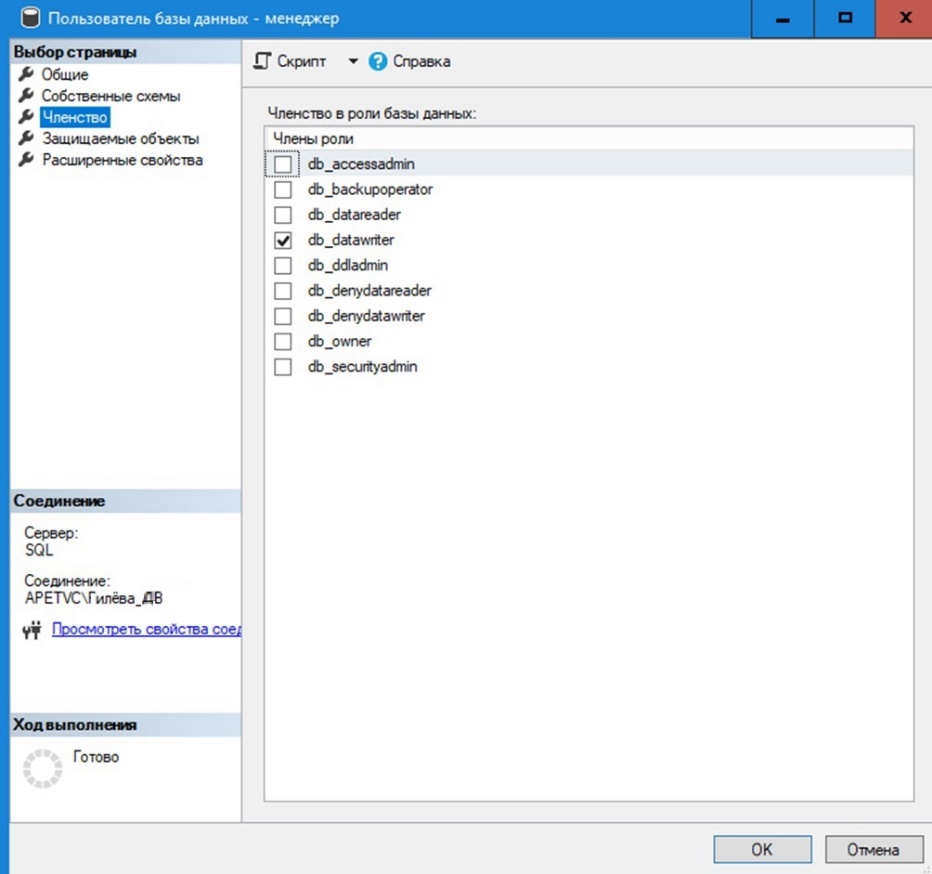


Рисунок 20 – Создание разграничения роли Менеджер

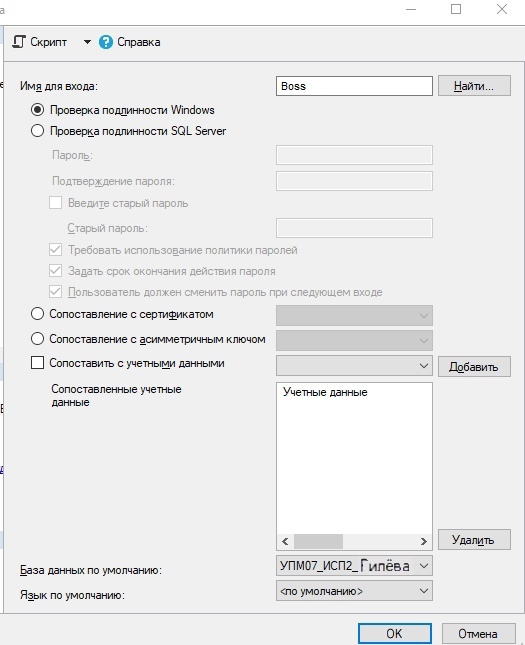


Рисунок 21 - Создание имени входа Директор

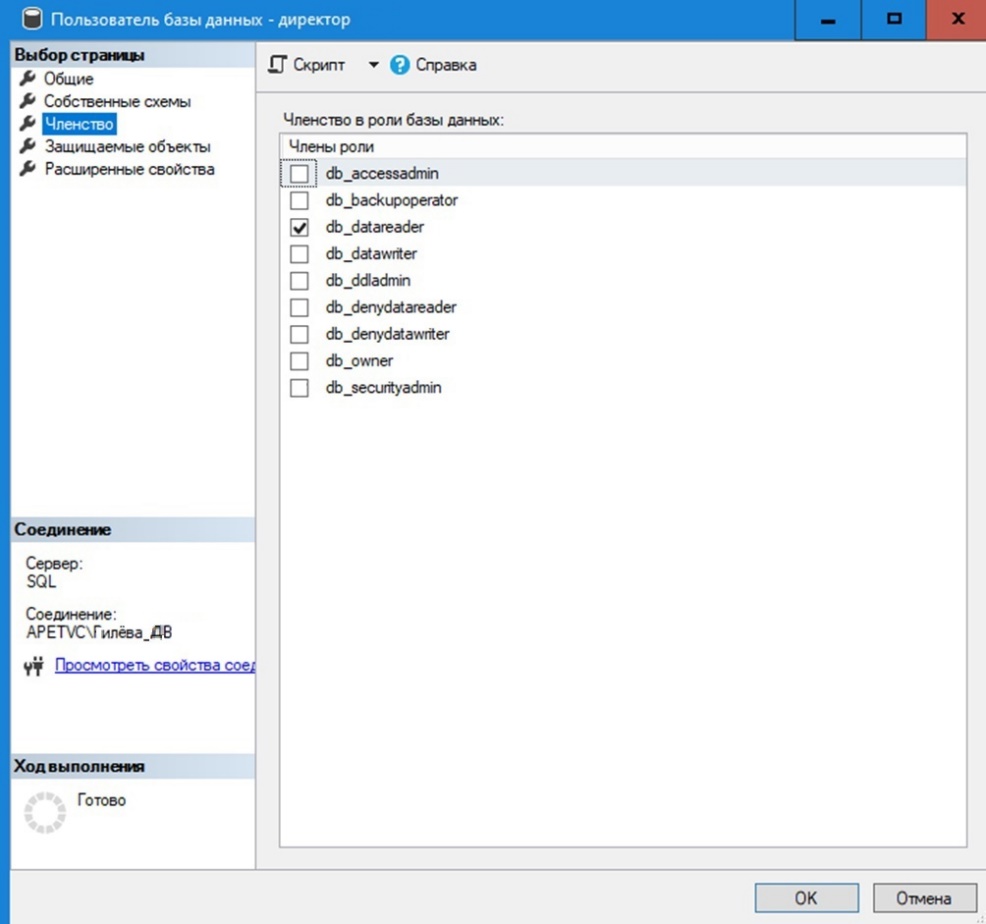


Рисунок 22 – Создание разграничения роли Пользователь

2.4.2 Настройка прав доступа к базе данных.

Используя SQL-команды в базе данных предоставить пользователю:

менеджер - полный доступ ко всем таблицам;

директор - просмотр основных таблиц, без возможности их редактирования;

администратор - привилегию SELECT на все таблицы базы данных и привилегию INSERT на основные таблицы.

2.4.3 Создание резервных копий базы данных. Восстановление базы данных из резервной копии

Выполнено резервное копирование базы данных SQL и клиентской части написанной на Microsoft Visual Studio С#.

Создание резервной копии серверной части, написанной на SQL Server:

1. нажатие правой кнопкой мыши по разработанной базе данных;
2. выбор пункта: создание резервной копии;
3. Создание имени резервной копии;
4. Нажатие на кнопку «ОК».

На рисунке 16 представлен вид создания резервной копии с её успешной выгрузкой в папку, указанную при создании.

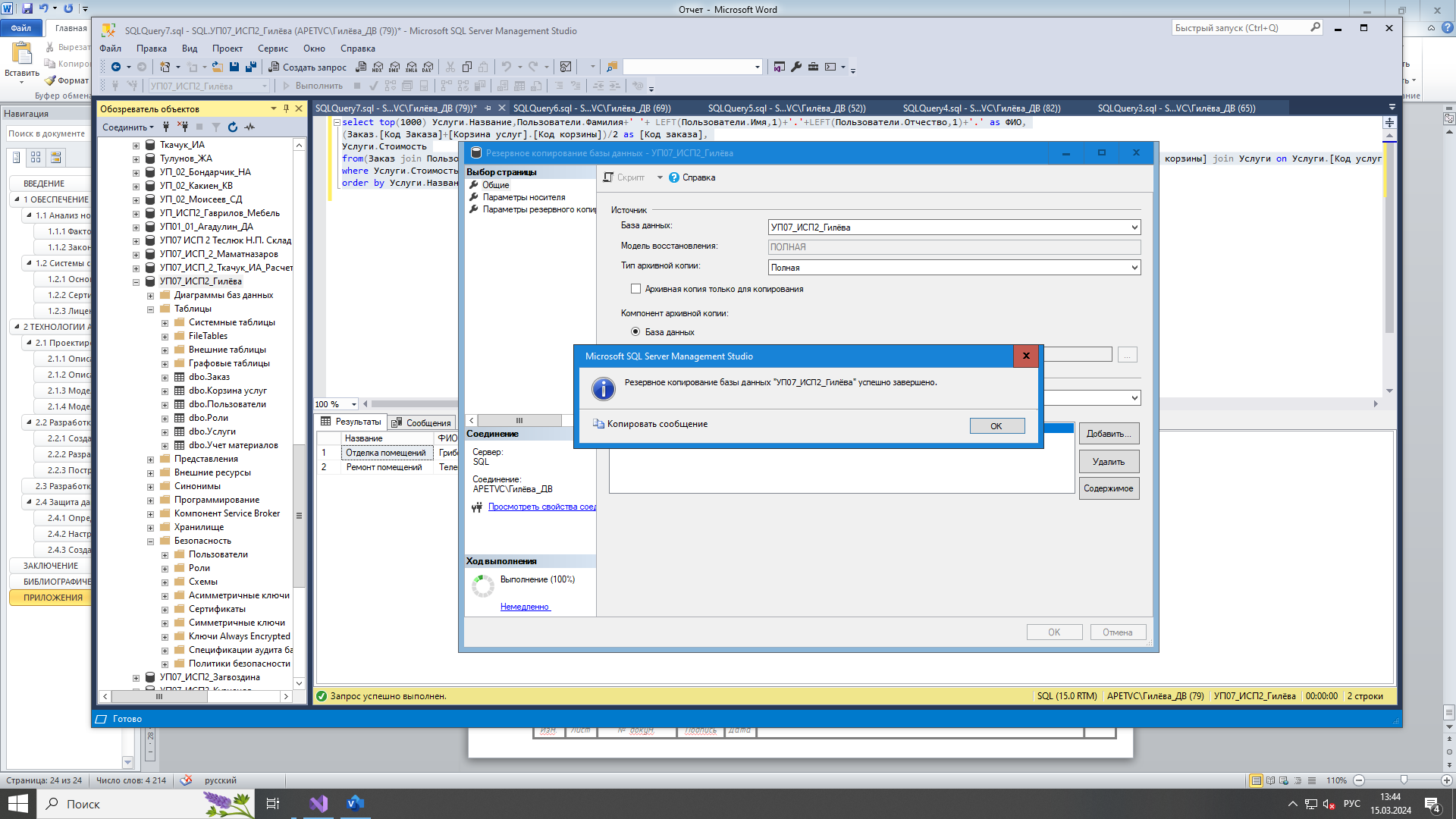


Рисунок 23 – создание резервной копии базы данных SQL Server

Создание резервной копии клиентского приложения разработанного в Microsoft Visual studio C#.

Резервное копирование для проекта Visual Studio C# представляет собой процесс сохранения копии файлов и данных проекта на внешний носитель, такой как внешний жесткий диск, облачное хранилище или USB-накопитель. Это позволяет защитить данные от потери в случае сбоя оборудования, ошибки пользователя или вирусных атак.

Для создания резервной копии проекта Visual Studio C# можно использовать встроенные инструменты программы или сторонние программы резервного копирования. Для стандартного резервного копирования подходит простая копия папки с проектом программы. Данная функция представлена на рисунке 24.

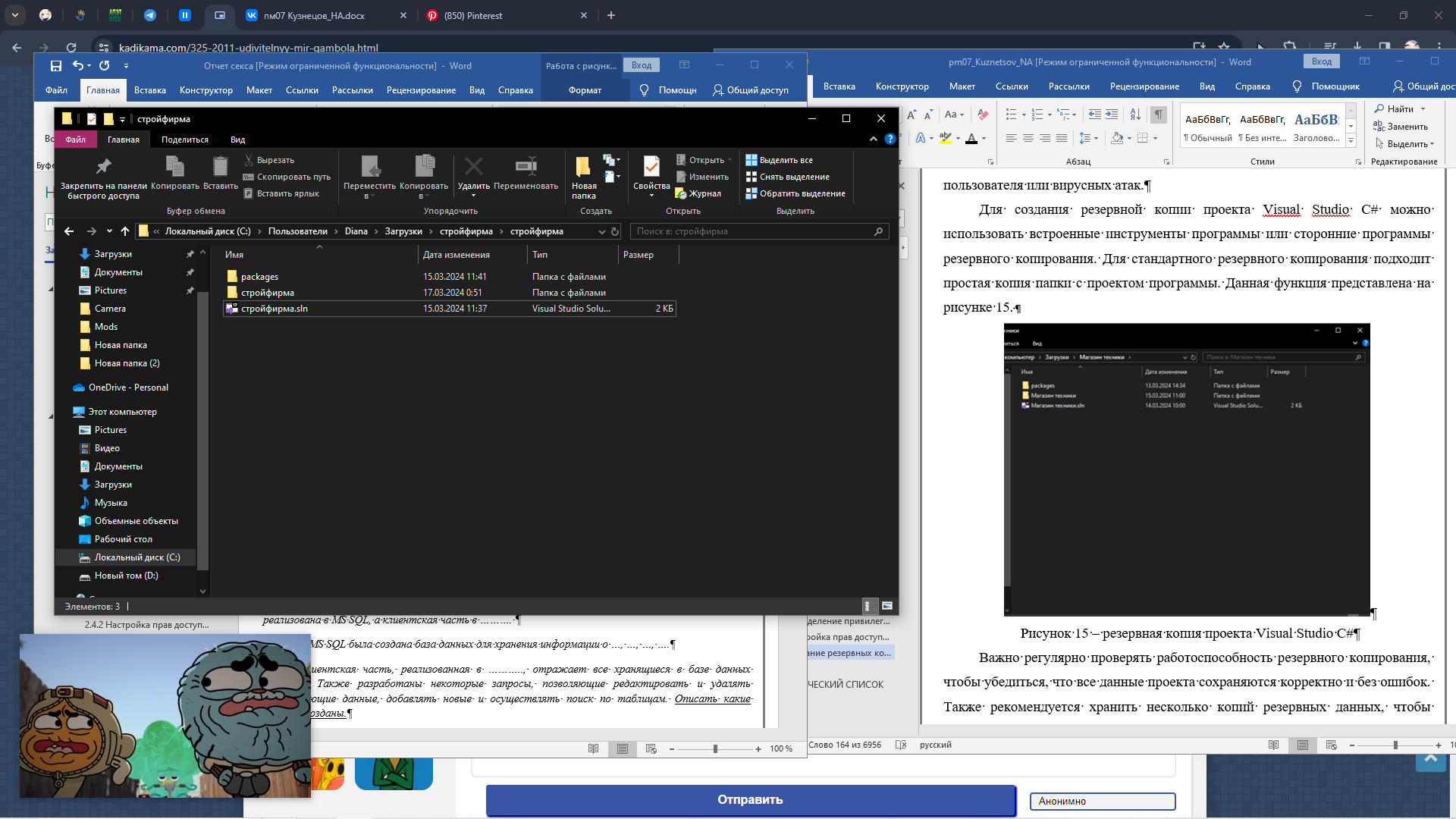


Рисунок 24 – резервная копия проекта Visual Studio С#

Важно регулярно проверять работоспособность резервного копирования, чтобы убедиться, что все данные проекта сохраняются корректно и без ошибок. Также рекомендуется хранить несколько копий резервных данных, чтобы обеспечить дополнительную защиту от потери информации.

Вывод по главе: По окончанию второй главы были спроектирована база данных MS SQL Server, Разработаны запросы для базы данных. Разработана клиентская часть в MS Visual Studio C#, настроены и разграничены права доступа, созданы резервные копии как базы данных, так и клиентского приложения.

ЗАКЛЮЧЕНИЕ

В результате учебной практики был проведен анализ предметной области, спроектирована и разработана информационная система "Система учета заказов и их выполнение в строительной фирме", серверная часть которой была реализована в MS SQL, а клиентская часть в Microsoft visual Studio.

В MS SQL Server была создана база данных для хранения информации о услугах и материалах, пользователях информационной системы, заказах.

Клиентская часть, реализованная в Microsoft Visual Studio, отражает все хранящиеся в базе данных таблицы. Также разработаны некоторые запросы, позволяющие редактировать и удалять существующие данные, добавлять новые и осуществлять поиск по таблицам. Разграничены права доступа и созданы резервные копии. Это позволяет сделать вывод, что поставленные цели и задачи выполнены в полном объеме.

Вся разработка была сохранена и выложена на общедоступном ресурсе GitHub по ссылке:

*.*

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Перлова О.Н., Ляпина О.П., Гусева А.В. «Проектирование и разработка информационных систем» - М.: Издательский центр «Академия», 2018. – 256 стр.
2. Соадминистрирование баз данных и серверов: учебник для студ. учреждений сред. проф. образования/ О.Н. Перлова, О.П. Ляпина. –М.: Издательский центр «Академия», 2018 – 304 с.
3. Учебник. Администрирование серверов с помощью управления на основе политик. Microsoft TechNet [Электронный доступ]. — Режим доступа: https://technet.microsoft.com/ru-ru/library/bb522659(v=sql.120). обращения 27.02.2018)
4. Электронно-библиотечная система «Лань» [Электронный доступ]. — Режим доступа: https://e.lanbook.com/ (Дата обращения: 27.02.2018 г.)
5. Образовательный портал INTUIT.RU [Электронный доступ]. — Режим доступа: https://www.intuit.ru/. (Дата обращения: 27.02.2018 г.);
6. Образовательный портал EDU [Электронный доступ]. — Режимдоступа: http://www.edu.ru /. (Дата обращения: 27.02.2018 г).

Интернет – ресурсы:

1. ИНТУИТ. Национальный открытый университет. Проектирование ИС. [Электронный ресурс] / http://www.intuit.ru/ - Электронные данные. – Режим доступа: http://www.intuit.ru/. Свободный. – Заглавие с экрана. – Яз. рус., анг. (06.12.2023)
2. Состав и структура АИС. [Электронный ресурс] / http://m60195.narod.ru. – Электронные данные. – Режим доступа: http://m60195.narod.ru/index/0-8. свободный. – Заглавие с экрана. – Яз. рус., англ. (06.12.2023)
3. Структура информационной системы. [Электронный ресурс] / http://domino.novsu.ac.ru/ – Электронные данные. – Режим доступа: http://domino.novsu.ac.ru/do/inf\_kult/gl32.htm. свободный. – Заглавие с экрана. – Яз. рус., англ. (07.12.2023)
4. Учебные материалы ВГУЭС. [Электронный ресурс] / http://abc.vvsu.ru/ – Электронные данные. – Режим доступа: http://abc.vvsu.ru/Books/inform\_tehnolog/page0010.asp. свободный. – Заглавие с экрана. – Яз. рус., англ. (07.12.2023)
5. Понятие информационной технологии (ИТ): определение, основные принципы и инструментарий. [Электронный ресурс] / http://cde.osu.ru/ - Электронные данные. – Режим доступа: http://cde.osu.ru/demoversion/course157/text/1.2.html. свободный. – Заглавие с экрана. – Яз. рус., англ. (07.12.2023)
6. Лекции по информационным технологиям / ИТ.doc. [Электронный ресурс] / http://www.studfiles.ru/ - Электронные данные. – Режим доступа: http://www.studfiles.ru/dir/cat32/subj1177/file9555/view96585/page2.html. свободный. – Заглавие с экрана. – Яз. рус., англ. (08.12.2023)
7. Современные информационные технологии и их классификация. [Электронный ресурс] / http://technologies.su/ - Электронные данные. – Режим доступа: http://technologies.su/klassifikaciya\_it. свободный. – Заглавие с экрана. – Яз. рус., англ. (08.12.2023)
8. Основные виды обработки данных. [Электронный ресурс] / http://dssp.petrsu.ru/ - Электронные данные. – Режим доступа: http://dssp.petrsu.ru/p/tutorial/informatics/chapter2/5/53.htm. свободный. – Заглавие с экрана. – Яз. рус., англ. (11.12.2023)
9. Системный анализ как основа проектирования информационных систем (Часть-I). [Электронный ресурс] / http://www.hbc.ru/ - Электронные данные. – Режим доступа: http://www.hbc.ru/news/analytics/217.html. свободный. – Заглавие с экрана. – Яз. рус., англ. (11.12.2023)
10. Методы сбора информации и инструменты анализа. [Электронный ресурс] / http://www.marketing.spb.ru/ - Электронные данные. – Режим доступа: http://www.marketing.spb.ru/libresearch/methods/collect\_and\_analysis.htm?printversion. свободный. – Заглавие с экрана. – Яз. рус., англ. (13.12.2023)
11. Проектирование информационных систем [Электронный ресурс] / http://pmn.narod.ru – Электронные данные. – Режим доступа: http://pmn.narod.ru/disciplins/dis\_cis.htm. свободный. – Заглавие с экрана. – Яз. рус., англ. (13.12.2023)
12. ИНТУИТ. Национальный открытый университет. Автоматизированное проектирование промышленных изделий. [Электронный ресурс] / http://www.intuit.ru/ - Электронные данные. – Режим доступа: http://www.intuit.ru/studies/courses/650/506/lecture/11501?page=2. свободный. – Заглавие с экрана. – Яз. рус., англ. (14.12.2023)
13. Мартишин, С. А. Базы данных практическое применение СУБД SQL и NOSQL-типа для проектирования информационных систем: учеб. пособие / С.А. Мартишин, В.Л. Симонов, М.В. Храпченко.-М.: ИД "ФОРУМ": ИНФРА-М, 2016.-368 с.