СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc162076839)

[1 ОБЕСПЕЧЕНИЕ КАЧЕСТВА И СЕРТИФИКАЦИЯ ИНФОРМАЦИОННЫХ СИСТЕМ 5](#_Toc162076840)

[1.1 Анализ нормативно-правовой документации, стандартов, регламентирующие качество программных средств 5](#_Toc162076841)

[1.1.1 Факторы качества программного обеспечения 5](#_Toc162076842)

[1.1.2 Законодательство Российской Федерации в области защиты информации 5](#_Toc162076843)

[1.2 Системы сертификации. Процедура сертификации 8](#_Toc162076844)

[1.2.1 Основные определения в области сертификации и лицензирования деятельности в сфере информатизации в Российской Федерации 8](#_Toc162076845)

[1.2.2 Сертификация средств и систем информатизации в Российской Федерации 9](#_Toc162076846)

[1.2.3 Лицензирование деятельности в сфере информатизации в Российской Федерации 11](#_Toc162076847)

[2 ТЕХНОЛОГИИ АДМИНИСТРИРОВАНИЯ СЕРВЕРОВ И БАЗ ДАННЫХ 12](#_Toc162076848)

[2.1 Проектирование базы данных 12](#_Toc162076849)

[2.1.1 Описание предметной области и основных требований, предъявляемых к информационной системе 12](#_Toc162076850)

[2.1.2 Описание групп пользователей информационной системы 13](#_Toc162076851)

[2.1.3 Моделирование диаграммы вариантов использования 13](#_Toc162076852)

[2.1.4 Моделирование данных, построение диаграммы «сущность-связь» 14](#_Toc162076853)

[2.2 Разработка и администрирование базы данных 16](#_Toc162076854)

[2.2.1 Создание физической модели данных 16](#_Toc162076855)

[2.2.2 Разработка серверной части базы данных в инструментальной среде 17](#_Toc162076856)

[2.3 Разработка клиентской части приложения в инструментальной среде 21](#_Toc162076857)

[2.4 Защита данных в хранилищах 44](#_Toc162076858)

[2.4.1 Определение привилегий пользователей с различными ролями. 44](#_Toc162076859)

[2.4.2 Настройка прав доступа к базе данных. 45](#_Toc162076860)

[2.4.3 Создание резервных копий базы данных. Восстановление базы данных из резервной копии 46](#_Toc162076861)

[ЗАКЛЮЧЕНИЕ 48](#_Toc162076862)

[ПРИЛОЖЕНИЕ А – Создание физической модели базы данных 49](#_Toc162076863)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 53](#_Toc162076864)

# ВВЕДЕНИЕ

Практическая подготовка в виде учебной практики – форма организации учебной деятельности в условиях выполнения определенных видов работ, связанных с будущей профессиональной деятельностью.

Целью практической подготовки в виде учебной практики является формирование, закрепление, развитие практических навыков и компетенций по профилю .

Основной задачей практической подготовки в виде учебной практики является, первоначально, анализ предметной области, а так же различных федеральных законов её проектирование путём построения диаграммы вариантов использования и логической модели данных. Далее разработка БД-приложения «Техцентра» со своей серверной и клиентской частью, защитой и разграничениями прав доступа, возможностью редактирования записей в хранилище данных в приложении, просмотра записей хранилища данных, а так же выборки значений согласно индивидуальному заданию.

# 1 ОБЕСПЕЧЕНИЕ КАЧЕСТВА И СЕРТИФИКАЦИЯ ИНФОРМАЦИОННЫХ СИСТЕМ

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

### 1.1.1 Факторы качества программного обеспечения

Согласно стандарту **ISO 9126:1-4 «**[Базовые принципы реализации метрологии и качества программного обеспечения](https://studbooks.net/2410707/informatika/bazovye_printsipy_realizatsii_metrologii_i_kachestva_programmnogo_obespecheniya)», качество программного обеспечения оценивается по следующим характеристикам:

1. **Функциональные возможности**: способность ИС обеспечивать функции, удовлетворяющие установленным потребностям заказчиков и пользователей.
2. **Надёжность**: способность ИС поддерживать определённую работоспособность в заданных условиях.
3. **Практичность**: свойство системы, характеризующееся сложностью её понимания, изучения и использования, а также привлекательностью для пользователя.
4. **Эффективность**: свойство системы обеспечивать требуемую производительность с учётом количества используемых вычислительных ресурсов.
5. **Сопровождаемость**: приспособленность системы к модификации и изменению конфигурации.
6. **Мобильность**: приспособленность системы к переносу из одной аппаратно-операционной среды в другую.

### 1.1.2 Законодательство Российской Федерации в области защиты информации

***«Закон РФ от 21.12.2023 № 2446-1 «О безопасности».***

Аннотация:

Безопасность - состояние защищенности жизненно важных интересов личности, общества и государства от внутренних и внешних угроз.

Жизненно важные интересы - совокупность потребностей, удовлетворение которых надежно обеспечивает существование и возможности прогрессивного развития личности, общества и государства.

К основным объектам безопасности относятся: личность - ее права и свободы; общество - его материальные и духовные ценности; государство - его конституционный строй, суверенитет и территориальная целостность.

***"Гражданский кодекс Российской Федерации (часть четвертая)" от 12.06.2023 N 230-ФЗ***

Аннотация:

На результаты интеллектуальной деятельности и приравненные к ним средства индивидуализации (результаты интеллектуальной деятельности и средства индивидуализации) признаются интеллектуальные права, которые включают исключительное право, являющееся имущественным правом, а в случаях, предусмотренных настоящим Кодексом, также личные неимущественные права и иные права (право следования, право доступа и другие).

Если произведение соавторов образует одно неразрывное целое, то ни один из соавторов не вправе без достаточных к тому оснований запретить использование произведения.

***Закон РФ от 01.02.2024 № 5485-1 «О государственной тайне».***

Аннотация:

Настоящий Закон регулирует отношения, связанные с отнесением сведений к государственной тайне, их засекречиванием или рассекречиванием и защитой государственной тайны в интересах обеспечения безопасности Российской Федерации.

***Федеральный закон от 30.08.2022 № 77-ФЗ «Об обязательном экземпляре документов».***

Аннотация:

Настоящий Федеральный закон определяет политику государства в области формирования обязательного экземпляра документов как ресурсной базы комплектования полного национального библиотечно-информационного фонда документов Российской Федерации и развития системы государственной библиографии, предусматривает обеспечение сохранности обязательного экземпляра документов, его общественное использование.

***Федеральный закон от 01.09.2023 № 63-ФЗ «Об электронной цифровой подписи».***

Настоящий Федеральный закон регулирует отношения в области использования электронных подписей при совершении гражданско-правовых сделок, оказании государственных и муниципальных услуг, исполнении государственных и муниципальных функций, при совершении иных юридически значимых действий, в том числе в случаях, установленных другими федеральными законами.

***Федеральный закон от 01.12.2023 № 126-ФЗ «О связи».***

Настоящий Федеральный закон регулирует отношения, связанные с созданием и эксплуатацией всех сетей связи и сооружений связи, использованием радиочастотного спектра, оказанием услуг электросвязи и почтовой связи на территории Российской Федерации и на находящихся под юрисдикцией Российской Федерации территориях.

***Федеральный закон от 25.07.2022 № 98-ФЗ «О коммерческой тайне».***

Аннотация:

Настоящий Федеральный закон регулирует отношения, связанные с установлением, изменением и прекращением режима коммерческой тайны в отношении информации, которая имеет действительную или потенциальную коммерческую ценность в силу неизвестности ее третьим лицам.

***Федеральный закон от 01.02.2024 № 149-ФЗ «Об информации, информационных технологиях и о защите информации».***

Аннотация:

1. Настоящий Федеральный закон регулирует отношения, возникающие при:

1) осуществлении права на поиск, получение, передачу, производство и распространение информации;

2) применении информационных технологий;

3) обеспечении защиты информации.

2. Положения настоящего Федерального закона не распространяются на отношения, возникающие при правовой охране результатов интеллектуальной деятельности и приравненных к ним средств индивидуализации, за исключением случаев, предусмотренных настоящим Федеральным законом.

## 1.2 Системы сертификации. Процедура сертификации

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

Сертификация — процедура, выполняемая третьей стороной, независимой от изгото­вителя (продавца) и потребителя продук­ции или услуг, по подтверждению соот­ветствия этих продукции или услуг уста­новленным требованиям.

Результатом выполнения процедуры сертифика­ции является так называемый сертификат соответ­ствия.

Сертификат соответствия — документ, выданный по правилам системы сертифи­кации для подтверждения соответствия сертифицированной продукции установленным требованиям.

Лицензирование – основным отличием процесса лицензирования от процесса сертификации является состав категорий, по отношению к которым они применяются. В процессе ли­цензирования фигурируют такие категории, как "деятельность" (подразумеваются виды или направления деятельности) и "субъект" (физическое лицо, предприя­тие, организация или иное юридическое лицо).

В соответствии с действующим законодатель­ством в Российской Федерации отдельные виды деятель­ности осуществляются предприятиями, организациями и учреждениями независимо от организационно-правовой формы, а также физическими лицами, осуществляющими предпринимательскую деятельность без образования юридического лица, на основании лицензии — специального разрешения органов, уполномоченных на ведение лицензи­рования.

Лицензия является официальным документом, ко­торый разрешает осуществление указанного в нем вида деятельности в течение установленного срока, а также определяет условия его осуществления.

Стандарт. Международная организация по стан­дартизации (ИСО) приняла следующее определение:

Стандарт - документ, составленный в сотрудни­честве и с согласия или общего одобрения всех заинтере­сованных в этом сторон, основанный на использовании обобщенных результатов науки, техники и практическо­го опыта, направленный на достижение оптимальной пользы для общества и утвержденный органом, зани­мающимся стандартизацией.

Национальный стандарт - документ, принятый национальным органом по стандартизации. Основная его функция согласно статусу этого органа или законам государства заключается в разработке и(или) опублико­вании национальных стандартов и(или) утверждении стандартов, подготовленных другими органами. Во всех странах мира национальные стандарты утверждаются на государственном уровне.

Международный стандарт - стандарт, принятый международным органом, занимающимся стандартиза­цией.

### 1.2.2 Сертификация средств и систем информатизации в Российской Федерации

В соответствии с действующими законодательными и нормативными документами выполнение работ по сер­тификации средств информатизации в данном направлении возложено на Госстандарт России. В 1994 году Гос­стандарт России ввел в действие нормативный документ "Номенклатура продукции и услуг, подлежащих обяза­тельной сертификации в Российской Федерации". Этот документ ежегодно пересматривается и уточняется с уче­том практики, условий торговли, производства и тен­денций научно-технического развития.

Указанным документом к продукции, подлежащей обязательной сертификации в рассматриваемом направ­лении, отнесены следующие средства информатизации:

* вычислительные машины и комплексы;
* персональные ЭВМ;
* устройства внешней памяти, ввода-вывода и от­ображения информации;
* устройства подготовки и телеобработки данных.

Поскольку основу сертификации по параметрам безопасности составляют общие требования к оборудо­ванию, остановимся подробнее на специфической для средств информатизации характеристике - электромаг­нитной совместимости.

Обеспечение электромагнитной совместимости за­ключается в выполнении требований по допустимым уровням электромагнитных помех, создаваемых функ­ционирующими средствами, и требований к помехоустой­чивости технических средств при воздействии внешних электромагнитных помех.

Невыполнение требований электромагнитной со­вместимости приводит к неэффективному использова­нию радиочастотного спектра, являющегося хотя и не расходуемым, но ограниченным ресурсом, к различным нарушениям в работе технических средств, а в ряде слу­чаев и к аварийным ситуациям.

Сертификация средств информатизации по требо­ваниям электромагнитной совместимости и параметрам безопасности возложена на Госстандарт России и прово­дится органами (центрами) сертификации, аккредито­ванными Госстандартом в рамках Системы сертифика­ции ГОСТ Р.

Для получения подобного сертификата изготови­тель или поставщик технических средств информатиза­ции должен обратиться в аккредитованный Госстандар­том России орган сертификации, представив комплект документов, определяемый правилами сертификации. Орган сертификации организует проведение соответ­ствующих испытаний (проверок) и при положительном результате испытаний выдает сертификат соответствия. В тексте сертификата указываются конкретные виды требований, по которым проведены испытания, и соот­ветствующие им нормативные документы.

### 1.2.3 Лицензирование деятельности в сфере информатизации в Российской Федерации

Общие принципы лицензирования видов деятель­ности в сфере информатизации России можно сформу­лировать следующим образом:

Целью лицензирования является защита интере­сов государства и граждан от неумышленного или сознательного некачественного выполнения работ, соответствующих определенным видам деятель­ности в сфере информатизации.

Виды деятельности в сфере информатизации, подлежащие лицензированию, а также органы, осуществляющие лицензирование конкретных ви­дов деятельности в различных областях информа­тизации, определены рядом нормативных докумен­тов.

Право на осуществление деятельности, подлежа­щей лицензированию, может получить субъект, от­вечающий определенным критериям, которые за­ранее определяются правилами проведения лицен­зирования и являющимися их неотъемлемой частью требованиями к предприятию-заявителю. Таким образом, субъектом лицензирования становится лишь то физическое или юридическое лицо, кото­рое представляет все необходимые и правильно оформленные документы и удовлетворяет соответ­ствующим требованиям.

# 2 ТЕХНОЛОГИИ АДМИНИСТРИРОВАНИЯ СЕРВЕРОВ И БАЗ ДАННЫХ

## 2.1 Проектирование базы данных

Проектирование базы данных — это процесс создания структурированного плана организации, хранения и управления данными для обеспечения целостности, согласованности и эффективности данных. Хорошо спроектированная база данных снижает избыточность данных, обеспечивает возможность повторного использования и упрощает управление данными . Проектирование хорошей базы данных предполагает использование лучших практик и методов, таких как моделирование данных, нормализация и моделирование отношений сущностей

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

Администрация службы техцентра заказала разработку информационной системы для отдела работы с клиентами. Система предназначена для обработки данных о комплектующих, о заказах на комплектующие, расценках по оказываемым услугам, о машинах и их обслуживании.

Приложение должно быть разработано с учетом специфики технологии «Клиент-сервер», содержать следующие возможности:

1. разграничение прав доступа к приложению.
2. редактирования записей хранилище данных в приложении (добавление, удаление, редактирование записей).
3. просмотра записей из хранилище данных.
4. выборки значений согласно индивидуальному заданию.
5. описать входную и выходную информацию.

Пользователь должен будет быть уверен в безопасности своей учетной записи и точно выставленных ему прав доступа, иметь возможность зарегистрироваться под своим логином и паролем в системе, не должен иметь никаких трудностей в освоении системы. Система должна быть интуитивно понятной и с удобным интерфейсом.

### 2.1.2 Описание групп пользователей информационной системы

Действующими лицами системы будут:

* менеджер
* администратор
* клиент

Администратор отвечает за работу непосредственно с самой базой данных, добавление новых таблиц, новых ролей, контролировать стабильную работу системы

Менеджер отвечает за добавление новых записей в базу данных, удаление и редактирование.

Клиент может ознакомится с ценами на услуги тех-центра, видеть сведения о машинах и их обслуживании

### 2.1.3 Моделирование диаграммы вариантов использования

Диаграмма вариантов использования-это графическое изображение возможных взаимодействий пользователя с системой. Диаграмма вариантов использования показывает различные варианты использования и различные типы пользователей, которые есть в системе. Диаграмма вариантов использования представлена на рисунке 1.



Рисунок 1 – диаграмма вариантов использрования

Администратор – добавление новых таблиц, добавление новых пользователей и выдачи им ролей.

Менеджер – добавление новых записей в БД, удаление записей в БД, просмотр всех таблиц.

Клиент – просмотр цен и услуг тех-центра.

### 2.1.4 Моделирование данных, построение диаграммы «сущность-связь»

Необходимо создать базу данных для работы с клиентами тех-центра.

Исходя из анализа предметной области, можно выделить два типа сущности (таблиц): Услуга и Клиент, которые связаны между собой типом отношений (Многие-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) Сделка.

**Описание отношений**: один покупатель, может неоднократно заказывать услуги, поэтому между объектами Клиент и Сделка имеется связь Один-ко-многим. Каждое наименование услуги может неоднократно участвовать в сделках, в результате между объектами Услуга и Сделка имеется связь Один-ко-многим.

**Описание сущностей:**

* К объекту Услуга относятся такие характеристики как: Название, цена, описание.
* К объекту Клиент относится: Фамилия, Имя, Отчество, Телефон.
* К объекту Сделка относится: Дата сделки.
* К объекту Сотрудники относится: Фамилия, Имя, Отчество, Телефон.

**Описание индексов:**

В таблице Услуги необходимо определить первичный ключ - Код товара, под которым можно понимать номер услуги.

В таблице Клиент необходимо определить первичный ключ - Код клиента, под которым можно понимать номер паспорта, ИНН или другой атрибут, однозначно определяющий каждого клиента.

В таблице Сделка первичным ключом будет поле - Код сделки, так как оно однозначно определяет дату покупки и другие элементы данных таблицы. В составной индекс таблицы Сделка (поля внешнего ключа) входят поля: Код сделки, Код клиента, Код услуги.

В таблице Сотрудники первичным ключом будет Код сотрудника, под которым можно понимать идентификационный номер каждого из сотрудников.

**Описание связей:**

Для обеспечения целостности данных между таблицами Клиент и Сделка по полю Код клиента тип связи определен как Один-ко-многим; между таблицами Услуги и Сделка, по полю Код услуги тип связи определен как Один-ко-многим.

Логическая модель данных базы данных представлена на рисунке 1.

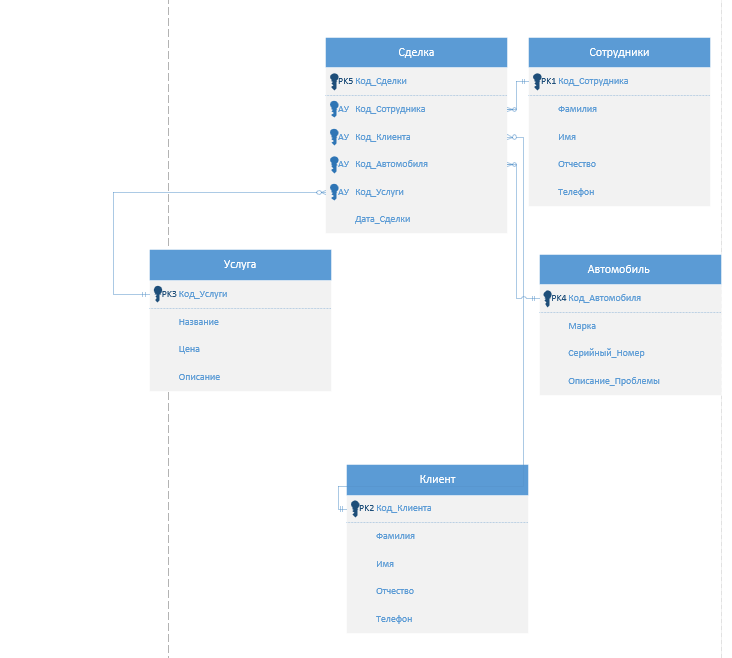
****

Рисунок 1 – Логическая модель данных

## 2.2 Разработка и администрирование базы данных

### 2.2.1 Создание физической модели данных

Физическая модель базы данных — это модель данных, которая определяет, каким образом представляются данные, и содержит все детали, необходимые СУБД для создания базы данных.

Для создания физической модели баз данных была использована СУБД Microsoft SQL Management Studio19.

Листинг программного кода представлен в приложении А.

В ходе выполнения заданий практической подготовки в виде учебной практики велась статистика состояния базы данных в процессе ее изменения. Результаты представлены в таблице 1.

Таблица 1 – Статистика состояния базы данных

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Название этапа | Размер базы | Нераспределенное пространство | Зарезервировано | Данные | Размер индекса | Не использовано |
| Создание БД | 108.00 MB | 97.60 MB | 2456 KB | 992 KB | 1072 KB | 392 KB |
| Создание таблиц | 108.00 MB | 97.48 MB | 2484 KB | 1024 KB | 1096 KB | 464 KB |
| Заполнение таблиц данными | 109.00 MB | 97.20 MB | 2472 KB | 1064 KB | 1128 KB | 680 KB |
| Создание пользователей | 109.00 MB | 97.16 MB | 2464 KB | 1081 KB | 1096 KB | 720 KB |
| Формирование запросов | 110.00 MB | 97.12 MB | 2458 KB | 1043 KB | 1073 KB | 801 KB |

### 2.2.2 Разработка серверной части базы данных в инструментальной среде

Листинг создания и заполнения таблиц представлен в приложении А.

Диаграмма схемы данных – разновидность блок-схемы, где показано, как разные «сущности»(люди, объекты, концепции и т.д.) связаны между собой представлена на рисунке 2

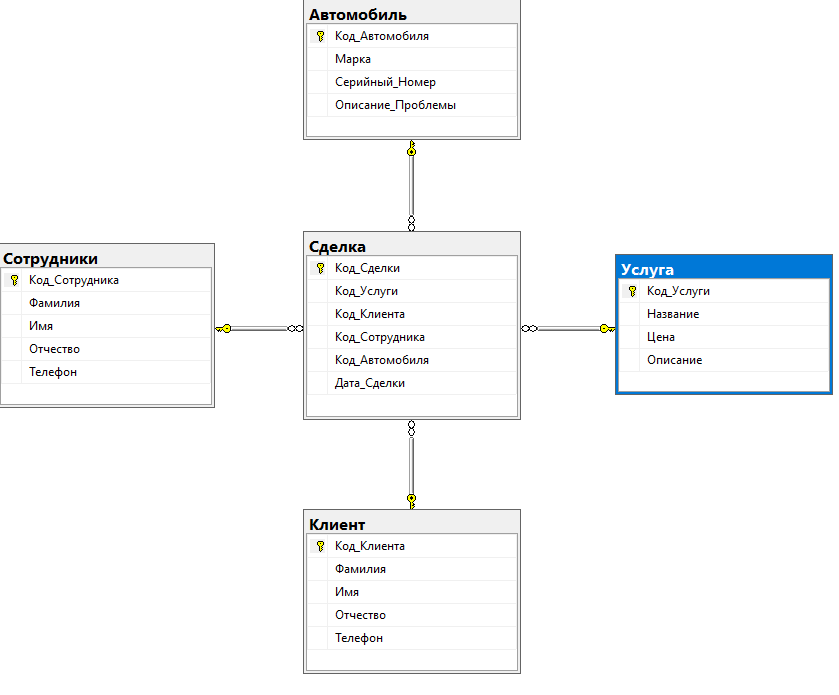


Рисунок 2 – Диаграмма схемы данных

2.2.3 Построение запросов к базе данных на языке SQL.

Необходимо, используя неструктурированный язык запросов Transact –SQL с помощью SQL кода создать запросы на выборку данных из таблиц базы данных ИС работы с клиентами тех-центра:

1. Вывести информацию о клиентах заказавших какую-либо услугу. Список упорядочить по дате сделки. Результат выполнения запроса представлен на рисунке 3. План выполнения на рисунке 4.

Код: SELECT Клиент.Фамилия,Клиент.Имя, Клиент.Отчество, Клиент.Телефон,Сделка.Дата\_Сделки,

Клиент.Фамилия+' '+Клиент.Имя+' '+Клиент.Отчество AS ФИО

FROM Сделка JOIN Клиент ON Клиент.Код\_Клиента = Сделка.Код\_Клиента ORDER BY Дата\_Сделки

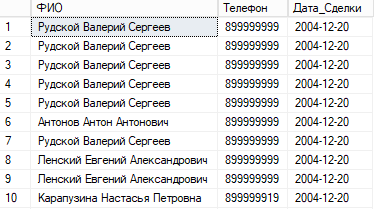


Рисунок 3 - Вывод информации о клиентах заказавших какую либо услугу. Упорядоченной по дате сделки

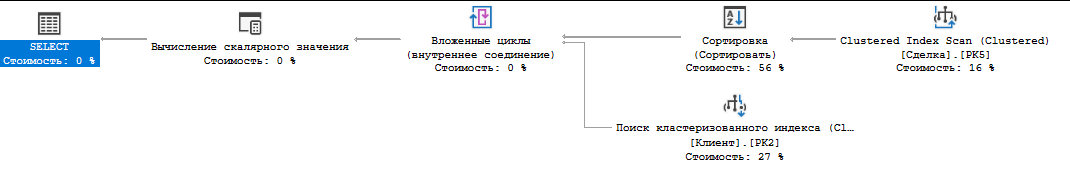


Рисунок 4 – План выполнения

1. Вывести информацию о сумме расходов каждого клиента. Результат выполнения запроса представлен на рисунке 5. На рисунке 6 план выполнения запроса.

Код: SELECT Клиент.Фамилия+' '+LEFT(Клиент.Имя,1)+' '+LEFT(Клиент.Отчество,1) AS ФИО, SUM(Услуга.Цена) as 'Сумма расходов'

FROM (Услуга JOIN Сделка ON Услуга.Код\_Услуги = Сделка.Код\_Услуги) JOIN Клиент ON Клиент.Код\_Клиента = Сделка.Код\_Клиента

GROUP BY Клиент.Фамилия+' '+LEFT(Клиент.Имя,1)+' '+LEFT(Клиент.Отчество,1)

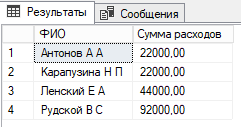


Рисунок 5 – Вывод информации о сумме расходов каждого клиента

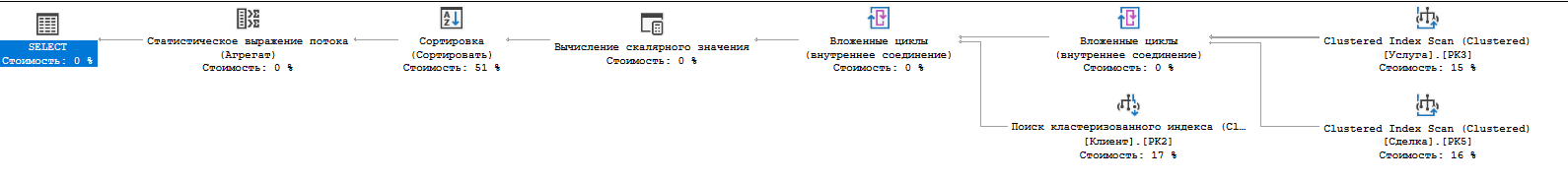


Рисунок 6 – План выполнения

1. Вывод минимальных цен на выполненние услуг тех – центра. Результат выполнения представлен на рисунке 7. На рисунке 8 план выполнения запроса.

Код: SELECT Клиент.Фамилия+' '+LEFT(Клиент.Имя,1)+' '+LEFT(Клиент.Отчество,1) AS ФИО, MIN(Услуга.Цена) as 'Минимальные цены за услугу'

FROM (Услуга JOIN Сделка ON Услуга.Код\_Услуги = Сделка.Код\_Услуги) JOIN Клиент ON Клиент.Код\_Клиента = Сделка.Код\_Клиента

GROUP BY Клиент.Фамилия+' '+LEFT(Клиент.Имя,1)+' '+LEFT(Клиент.Отчество,1)



Рисунок 7 - Вывод минимальных цен на выполнение услуг тех – центра

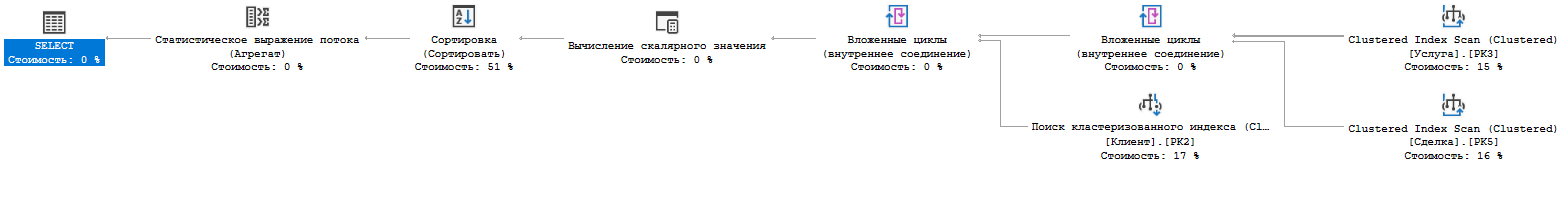


Рисунок 8 – План выполнения

1. Вывод списка услуг проданных на сумму менее 100000 рублей. Результат выполнения представлен на рисунке 9. На рисунке 10 план выполнения запроса.

Код: SELECT Услуга.Название, SUM(Услуга.Цена) as Стоимость

FROM (Услуга JOIN Сделка ON Услуга.Код\_Услуги = Сделка.Код\_Услуги)

WHERE Услуга.Цена<100000

group by Услуга.Название

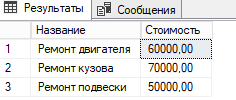


Рисунок 9 - Вывод списка услуг проданных на сумму менее 100000 рублей

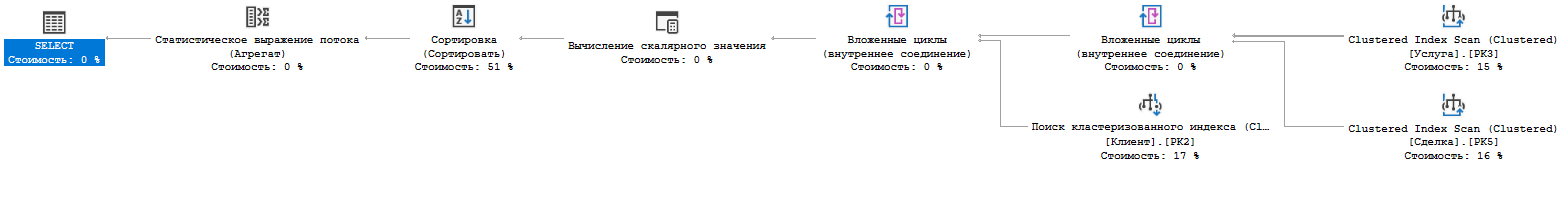


Рисунок 10 – План выполнения

1. Вывод данных о том, на какую сумму было продано товаров за март месяц. Результат выполения представлен на рисунке 11. На рисунке 12 план выполнения запроса.

Код: SELECT Услуга.Название, SUM(Услуга.Цена) as Стоимость,

MONTH (Сделка.Дата\_Сделки) as Месяц

FROM (Услуга JOIN Сделка ON Услуга.Код\_Услуги = Сделка.Код\_Услуги)

WHERE MONTH(Сделка.Дата\_Сделки)=3

group by Название,MONTH(Сделка.Дата\_Сделки)

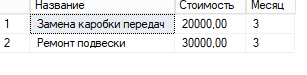


Рисунок 11 - Вывод данных о том, на какую сумму было продано товаров за март месяц

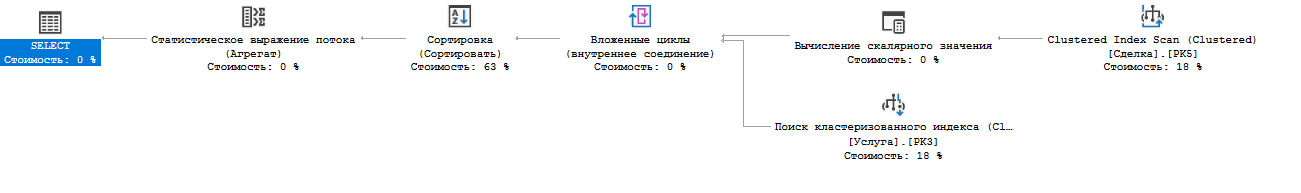


Рисунок 12 – План выполнения

## 2.3 Разработка клиентской части приложения в инструментальной среде

Для интегрированной базы данных создать интерфейс пользователя (форму авторизации, формы навигации по приложению и др.).

Форма авторизации представлена на рисунке 8

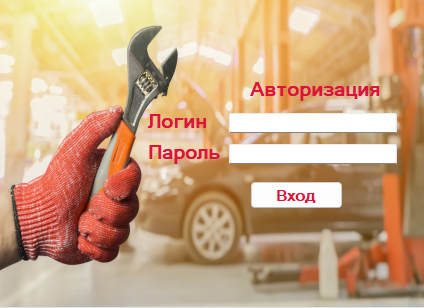


Рисунок 13 - Форма авторизации

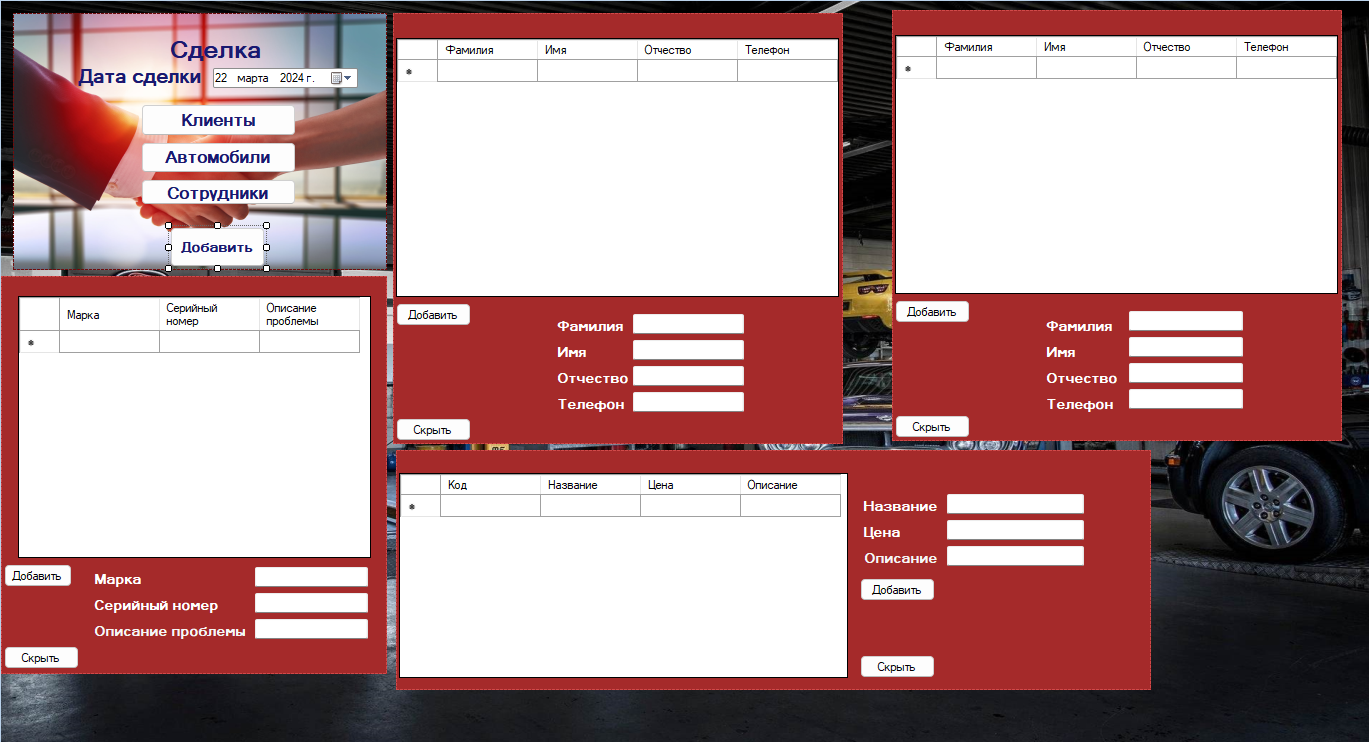
Форма «Менеджер» представлена на рисунке 9

Рисунок 14 – Форма «Менеджер»

Форма «Администратор представлена на рисунке 10

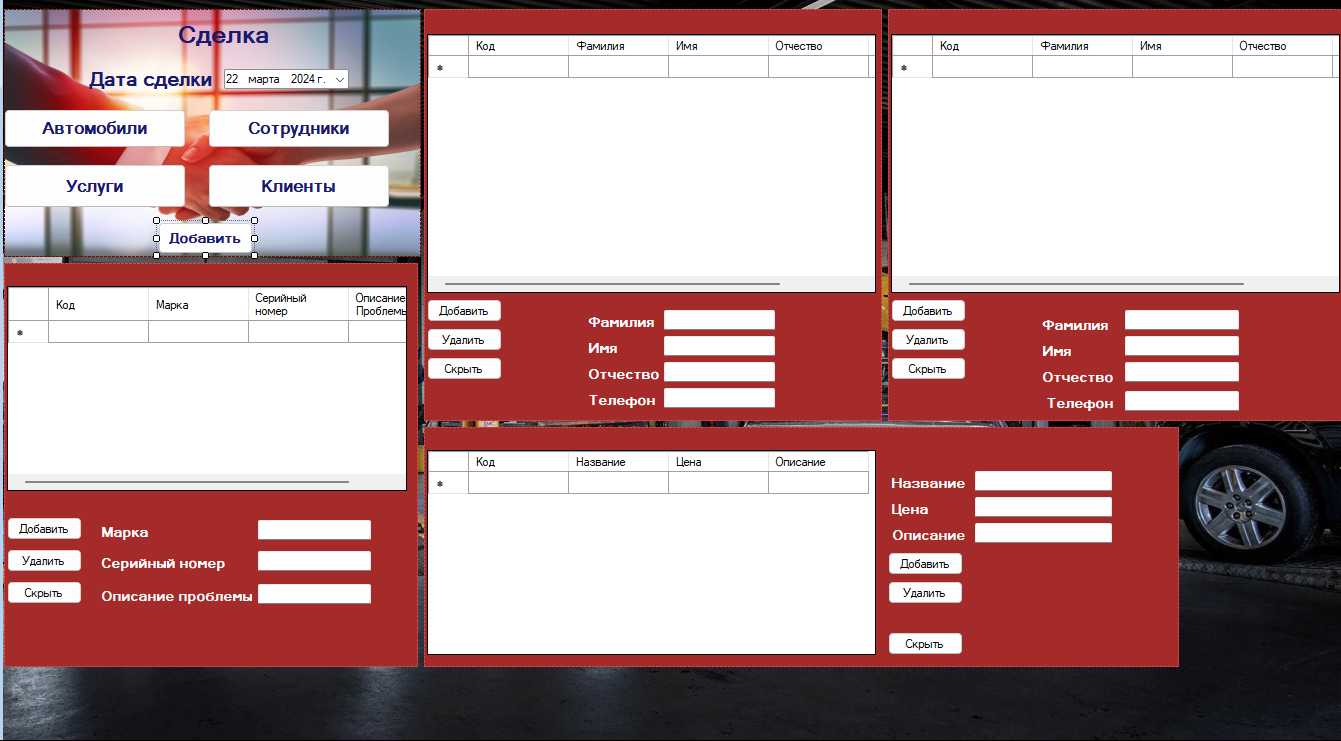


Рисунок 16 – Форма «Администратор»

Форма «Директор» представлена на рисунке 11

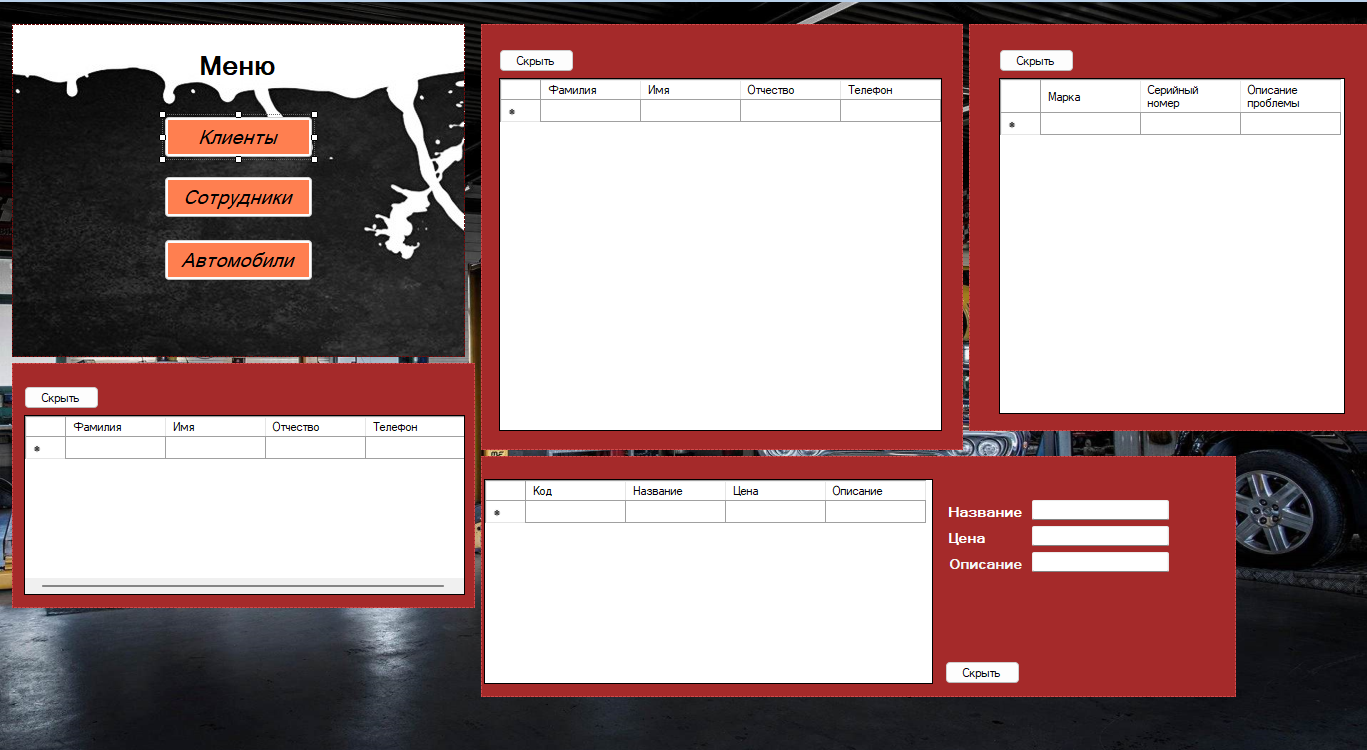


Рисунок 15 – Форма «Директор»

Все разработанные формы продемонстрировать на рисунках в отчете, проиллюстрировав его рисунками и таблицами.

Таблица 1 – Событийно-управляемые процедуры формы авторизации

| Объект | Событийно-управляемые процедуры | Код |
| --- | --- | --- |
| Button1 | private void button1\_Click(object sender, EventArgs e) | private void button1\_Click(object sender, EventArgs e)  {  if (textBox1.Text == "Менеджер" & textBox2.Text == "12345")  {  Form Form2 = new Form2();  Form2.Show();  this.Hide();  }  else if (textBox1.Text == "Директор" & textBox2.Text == "123456")  {  Form Form4 = new Form4();  Form4.Show();  this.Hide();  }  else if (textBox1.Text == "Администратор" & textBox2.Text == "1234")  {  Form Form3 = new Form3();  Form3.Show();  this.Hide();  }  else  {  MessageBox.Show("Неверный логин или пароль");  }  }  }  } |

Таблица 2 – Событийно-управляемые процедуры формы «Менеджер»

| Объекты | Свойства | Значения |
| --- | --- | --- |
| Button1 | private void button1\_Click(object sender, EventArgs e) | panel3.Show(); |
| Button2 | private void button2\_Click(object sender, EventArgs e) | panel1.Show(); |
| Button6 |  | panel5.Show(); |
| Button7 | private void button7\_Click(object sender, EventArgs e) | panel3.Hide(); |
| Button8 | private void button8\_Click(object sender, EventArgs e) | panel1.Hide(); |
| Button9 | private void button9\_Click(object sender, EventArgs e) | panel5.Hide(); |
| Button15 | private void button15\_Click(object sender, EventArgs e) | panel4.Hide(); |
| Form2\_Load | private void Form2\_Load(object sender, EventArgs e) | // TODO: данная строка кода позволяет загрузить данные в таблицу "andrei.Услуга". При необходимости она может быть перемещена или удалена.  this.услугаTableAdapter.Fill(this.andrei.Услуга);  // TODO: данная строка кода позволяет загрузить данные в таблицу "andrei.Сотрудники". При необходимости она может быть перемещена или удалена.  this.сотрудникиTableAdapter.Fill(this.andrei.Сотрудники);  // TODO: данная строка кода позволяет загрузить данные в таблицу "andrei.Клиент". При необходимости она может быть перемещена или удалена.  this.клиентTableAdapter.Fill(this.andrei.Клиент);  // TODO: данная строка кода позволяет загрузить данные в таблицу "andrei.Автомобиль". При необходимости она может быть перемещена или удалена.  this.автомобильTableAdapter.Fill(this.andrei.Автомобиль); |
| Button3 | private void button3\_Click(object sender, EventArgs e) | DB db = new DB();  db.openconn();  if (textBox1.Text == "" || textBox2.Text == "" || textBox3.Text == "" || textBox4.Text == "")  {  MessageBox.Show("Ошибка! Поля не заполнены");  }  else  {  SqlCommand command = new SqlCommand("INSERT INTO Сотрудники(Фамилия, Имя, Отчество,Телефон) VALUES(@Фамилия, @Имя, @Отчество,@Телефон)",db.getconn()) ;  command.Parameters.AddWithValue("@Фамилия", textBox1.Text);  command.Parameters.AddWithValue("@Имя", textBox2.Text);  command.Parameters.AddWithValue("@Отчество", textBox3.Text);  command.Parameters.AddWithValue("@Телефон", textBox4.Text);  command.ExecuteNonQuery();  MessageBox.Show("Сотрудник добавлен");  db.closeconn();  this.сотрудникиTableAdapter.Fill(this.andrei.Сотрудники); |
| Button4 | private void button4\_Click(object sender, EventArgs e) | DB db = new DB();  db.openconn();  if (textBox5.Text == "" || textBox6.Text == "" || textBox7.Text == "" || textBox8.Text == "")  {  MessageBox.Show("Ошибка! Поля не заполнены");  }  else  {  SqlCommand command = new SqlCommand("INSERT INTO Клиент(Фамилия, Имя, Отчество,Телефон) VALUES(@Фамилия, @Имя, @Отчество,@Телефон)", db.getconn());  command.Parameters.AddWithValue("@Фамилия", textBox5.Text);  command.Parameters.AddWithValue("@Имя", textBox6.Text);  command.Parameters.AddWithValue("@Отчество", textBox7.Text);  command.Parameters.AddWithValue("@Телефон", textBox8.Text);  command.ExecuteNonQuery();  MessageBox.Show("Клиент добавлен");  db.closeconn();  this.клиентTableAdapter.Fill(this.andrei.Клиент);  }  } |
| Button 13 | private void button13\_Click(object sender, EventArgs e) | DB db = new DB();  db.openconn();  DateTime selectedDateTime = dateTimePicker1.Value;  SqlCommand command = new SqlCommand("INSERT INTO Сделка(Код\_Сотрудника,Код\_Клиента,Код\_Автомобиля,Код\_Услуги, Дата\_Сделки ) VALUES(@Код\_Сотрудника,@Код\_Клиента,@Код\_Автомобиля,@Код\_Услуги,@Дата\_Сделки)", db.getconn());  command.Parameters.AddWithValue("@Код\_Сотрудника", dataGridView1[0, dataGridView1.CurrentRow.Index].Value.ToString());  command.Parameters.AddWithValue("@Код\_Клиента", dataGridView2[0, dataGridView2.CurrentRow.Index].Value.ToString());  command.Parameters.AddWithValue("@Код\_Автомобиля", dataGridView3[0, dataGridView3.CurrentRow.Index].Value.ToString());  command.Parameters.AddWithValue("@Код\_Услуги", dataGridView4[0, dataGridView4.CurrentRow.Index].Value.ToString());  command.Parameters.Add("@Дата\_Сделки", SqlDbType.Date).Value = selectedDateTime;  command.ExecuteNonQuery();  MessageBox.Show("Сделка добавлена");  db.closeconn(); |
| Button5 | private void button5\_Click(object sender, EventArgs e) | DB db = new DB();  db.openconn();  if (textBox9.Text == "" || textBox10.Text == "" || textBox11.Text == "")  {  MessageBox.Show("Ошибка! Поля не заполнены");  }  else  {  SqlCommand command = new SqlCommand("INSERT INTO Автомобиль(Марка, Серийный\_Номер, Описание\_Проблемы) VALUES(@Марка, @Серийный\_Номер, @Описание\_Проблемы)", db.getconn());  command.Parameters.AddWithValue("@Марка", textBox11.Text);  command.Parameters.AddWithValue("@Серийный\_Номер", textBox9.Text);  command.Parameters.AddWithValue("@Описание\_Проблемы", textBox10.Text);  command.ExecuteNonQuery();  MessageBox.Show("Автомобиль добавлен");  db.closeconn();  this.автомобильTableAdapter.Fill(this.andrei.Автомобиль);  } |
| Button 16 | private void button16\_Click(object sender, EventArgs e) | DB db = new DB();  db.openconn();  if (textBox5.Text == "" || textBox6.Text == "" || textBox7.Text == "" || textBox8.Text == "")  {  MessageBox.Show("Ошибка! Поля не заполнены");  }  else  {  SqlCommand command = new SqlCommand("INSERT INTO Услуга(Название, Цена, Описание) VALUES(@Название, @Цена, @Описание)", db.getconn());  command.Parameters.AddWithValue("@Название", textBox14.Text);  command.Parameters.AddWithValue("@Цена", textBox13.Text);  command.Parameters.AddWithValue("@Описание", textBox12.Text);  command.ExecuteNonQuery();  MessageBox.Show("Услуга добавлена");  db.closeconn();  this.услугаTableAdapter.Fill(this.andrei.Услуга);  } |

Таблица 3 – Событийно-управляемые процедуры формы «Администратор»

| Объекты | Свойства | Значения |
| --- | --- | --- |
| Button1 | private void button1\_Click(object sender, EventArgs e) | panel3.Show(); |
| Button2 | private void button2\_Click(object sender, EventArgs e) | panel1.Show(); |
| Button6 | private void button6\_Click(object sender, EventArgs e) | panel5.Show(); |
| Button7 | private void button7\_Click(object sender, EventArgs e) | panel3.Hide(); |
| Button8 | private void button8\_Click(object sender, EventArgs e) | panel1.Hide(); |
| Button9 | private void button9\_Click(object sender, EventArgs e) | panel5.Hide(); |
| Button15 | private void button5\_Click(object sender, EventArgs e) | panel4.Hide(); |
| Form3\_Load | private void Form3\_Load(object sender, EventArgs e) | // TODO: данная строка кода позволяет загрузить данные в таблицу "andrei.Услуга". При необходимости она может быть перемещена или удалена.  this.услугаTableAdapter.Fill(this.andrei.Услуга);  // TODO: данная строка кода позволяет загрузить данные в таблицу "andrei.Сотрудники". При необходимости она может быть перемещена или удалена.  this.сотрудникиTableAdapter.Fill(this.andrei.Сотрудники);  // TODO: данная строка кода позволяет загрузить данные в таблицу "andrei.Клиент". При необходимости она может быть перемещена или удалена.  this.клиентTableAdapter.Fill(this.andrei.Клиент);  // TODO: данная строка кода позволяет загрузить данные в таблицу "andrei.Автомобиль". При необходимости она может быть перемещена или удалена.  this.автомобильTableAdapter.Fill(this.andrei.Автомобиль); |
| Button3 | private void button3\_Click(object sender, EventArgs e) | DB db = new DB();  db.openconn();  if (textBox1.Text == "" || textBox2.Text == "" || textBox3.Text == "" || textBox4.Text == "")  {  MessageBox.Show("Ошибка! Поля не заполнены");  }  else  {  SqlCommand command = new SqlCommand("INSERT INTO Сотрудники(Фамилия, Имя, Отчество,Телефон) VALUES(@Фамилия, @Имя, @Отчество,@Телефон)",db.getconn()) ;  command.Parameters.AddWithValue("@Фамилия", textBox1.Text);  command.Parameters.AddWithValue("@Имя", textBox2.Text);  command.Parameters.AddWithValue("@Отчество", textBox3.Text);  command.Parameters.AddWithValue("@Телефон", textBox4.Text);  command.ExecuteNonQuery();  MessageBox.Show("Сотрудник добавлен");  db.closeconn();  this.сотрудникиTableAdapter.Fill(this.andrei.Сотрудники); |
| Button4 | private void button4\_Click(object sender, EventArgs e) | DB db = new DB();  db.openconn();  if (textBox5.Text == "" || textBox6.Text == "" || textBox7.Text == "" || textBox8.Text == "")  {  MessageBox.Show("Ошибка! Поля не заполнены");  }  else  {  SqlCommand command = new SqlCommand("INSERT INTO Клиент(Фамилия, Имя, Отчество,Телефон) VALUES(@Фамилия, @Имя, @Отчество,@Телефон)", db.getconn());  command.Parameters.AddWithValue("@Фамилия", textBox5.Text);  command.Parameters.AddWithValue("@Имя", textBox6.Text);  command.Parameters.AddWithValue("@Отчество", textBox7.Text);  command.Parameters.AddWithValue("@Телефон", textBox8.Text);  command.ExecuteNonQuery();  MessageBox.Show("Клиент добавлен");  db.closeconn();  this.клиентTableAdapter.Fill(this.andrei.Клиент);  }  } |
| Button5 | private void button5\_Click(object sender, EventArgs e) | DB db = new DB();  db.openconn();  if (textBox9.Text == "" || textBox10.Text == "" || textBox11.Text == "")  {  MessageBox.Show("Ошибка! Поля не заполнены");  }  else  {  SqlCommand command = new SqlCommand("INSERT INTO Автомобиль(Марка, Серийный\_Номер, Описание\_Проблемы) VALUES(@Марка, @Серийный\_Номер, @Описание\_Проблемы)", db.getconn());  command.Parameters.AddWithValue("@Марка", textBox11.Text);  command.Parameters.AddWithValue("@Серийный\_Номер", textBox9.Text);  command.Parameters.AddWithValue("@Описание\_Проблемы", textBox10.Text);  command.ExecuteNonQuery();  MessageBox.Show("Автомобиль добавлен");  db.closeconn();  this.автомобильTableAdapter.Fill(this.andrei.Автомобиль);  } |
| Button 16 | private void button16\_Click(object sender, EventArgs e) | DB db = new DB();  db.openconn();  if (textBox5.Text == "" || textBox6.Text == "" || textBox7.Text == "" || textBox8.Text == "")  {  MessageBox.Show("Ошибка! Поля не заполнены");  }  else  {  SqlCommand command = new SqlCommand("INSERT INTO Услуга(Название, Цена, Описание) VALUES(@Название, @Цена, @Описание)", db.getconn());  command.Parameters.AddWithValue("@Название", textBox14.Text);  command.Parameters.AddWithValue("@Цена", textBox13.Text);  command.Parameters.AddWithValue("@Описание", textBox12.Text);  command.ExecuteNonQuery();  MessageBox.Show("Услуга добавлена");  db.closeconn();  this.услугаTableAdapter.Fill(this.andrei.Услуга);  } |
| Button10 | private void button10\_Click(object sender, EventArgs e) | DB db = new DB();  db.openconn();  string query = ("DELETE FROM Сделка WHERE Код\_Сотрудника=@id");  SqlCommand command = new SqlCommand(query, db.getconn());  command.Parameters.AddWithValue("@id", dataGridView1[0, dataGridView1.CurrentRow.Index].Value.ToString());  command.ExecuteNonQuery();  query = ("DELETE FROM Сотрудники WHERE Код\_Сотрудника=@id");  command =new SqlCommand(query,db.getconn());  command.Parameters.AddWithValue("@id", dataGridView1[0, dataGridView1.CurrentRow.Index].Value.ToString());  command.ExecuteNonQuery();  MessageBox.Show("Сотрудник удален");  db.closeconn();  this.сотрудникиTableAdapter.Fill(this.andrei.Сотрудники); |
| Button11 | private void button11\_Click(object sender, EventArgs e) | DB db = new DB();  db.openconn();  string query = ("DELETE FROM Сделка WHERE Код\_Клиента=@id");  SqlCommand command = new SqlCommand(query, db.getconn());  command.Parameters.AddWithValue("@id", dataGridView2[0, dataGridView2.CurrentRow.Index].Value.ToString());  command.ExecuteNonQuery();  query = ("DELETE FROM Клиент WHERE Код\_Клиента=@id");  command = new SqlCommand(query, db.getconn());  command.Parameters.AddWithValue("@id", dataGridView2[0, dataGridView2.CurrentRow.Index].Value.ToString());  command.ExecuteNonQuery();  MessageBox.Show("Клиент удален");  db.closeconn();  this.клиентTableAdapter.Fill(this.andrei.Клиент); |
| Button12 | private void button12\_Click(object sender, EventArgs e) | DB db = new DB();  db.openconn();  string query = ("DELETE FROM Сделка WHERE Код\_Автомобиля=@id");  SqlCommand command = new SqlCommand(query, db.getconn());  command.Parameters.AddWithValue("@id", dataGridView3[0, dataGridView3.CurrentRow.Index].Value.ToString());  command.ExecuteNonQuery();  query = ("DELETE FROM Автомобиль WHERE Код\_Автомобиля=@id");  command = new SqlCommand(query, db.getconn());  command.Parameters.AddWithValue("@id", dataGridView3[0, dataGridView3.CurrentRow.Index].Value.ToString());  command.ExecuteNonQuery();  MessageBox.Show("Клиент удален");  db.closeconn();  this.автомобильTableAdapter.Fill(this.andrei.Автомобиль); |
| Button13 | private void button13\_Click(object sender, EventArgs e) | DB db = new DB();  db.openconn();  DateTime selectedDateTime = dateTimePicker1.Value;  SqlCommand command = new SqlCommand("INSERT INTO Сделка(Код\_Сотрудника,Код\_Клиента,Код\_Автомобиля,Код\_Услуги, Дата\_Сделки ) VALUES(@Код\_Сотрудника,@Код\_Клиента,@Код\_Автомобиля,@Код\_Услуги,@Дата\_Сделки)", db.getconn());  command.Parameters.AddWithValue("@Код\_Сотрудника", dataGridView1[0, dataGridView1.CurrentRow.Index].Value.ToString());  command.Parameters.AddWithValue("@Код\_Клиента", dataGridView2[0, dataGridView2.CurrentRow.Index].Value.ToString());  command.Parameters.AddWithValue("@Код\_Автомобиля", dataGridView3[0, dataGridView3.CurrentRow.Index].Value.ToString());  command.Parameters.AddWithValue("@Код\_Услуги", dataGridView4[0, dataGridView4.CurrentRow.Index].Value.ToString());  command.Parameters.Add("@Дата\_Сделки", SqlDbType.Date).Value = selectedDateTime;  command.ExecuteNonQuery();  MessageBox.Show("Сделка добавлена");  db.closeconn(); |
| Buttom14 | private void button14\_Click(object sender, EventArgs e) | DB db = new DB();  db.openconn();  string query = ("DELETE FROM Сделка WHERE Код\_Услуги=@id");  SqlCommand command = new SqlCommand(query, db.getconn());  command.Parameters.AddWithValue("@id", dataGridView3[0, dataGridView3.CurrentRow.Index].Value.ToString());  command.ExecuteNonQuery();  query = ("DELETE FROM Услуга WHERE Код\_Услуги=@id");  command = new SqlCommand(query, db.getconn());  command.Parameters.AddWithValue("@id", dataGridView4[0, dataGridView4.CurrentRow.Index].Value.ToString());  command.ExecuteNonQuery();  MessageBox.Show("Услуга удалена");  db.closeconn();  this.услугаTableAdapter.Fill(this.andrei.Услуга); |

Таблица 4 – Событийно-управляемые процедуры формы «Директор»

| Объекты | Свойства | Значения |
| --- | --- | --- |
| Button1 | private void button1\_Click(object sender, EventArgs e) | panel3.Show(); |
| Button2 | private void button2\_Click(object sender, EventArgs e) | panel1.Show(); |
| Button6 | private void button6\_Click(object sender, EventArgs e) | panel5.Show(); |
| Button7 | private void button7\_Click(object sender, EventArgs e) | panel3.Hide(); |
| Button8 | private void button8\_Click(object sender, EventArgs e) | panel1.Hide(); |
| Button9 | private void button9\_Click(object sender, EventArgs e) | panel5.Hide(); |
| Button15 | private void button15\_Click(object sender, EventArgs e) | panel4.Hide(); |
| Form4\_Load | private void Form4\_Load(object sender, EventArgs e) | // TODO: данная строка кода позволяет загрузить данные в таблицу "andrei.Услуга". При необходимости она может быть перемещена или удалена.  this.услугаTableAdapter.Fill(this.andrei.Услуга);  // TODO: данная строка кода позволяет загрузить данные в таблицу "andrei.Сотрудники". При необходимости она может быть перемещена или удалена.  this.сотрудникиTableAdapter.Fill(this.andrei.Сотрудники);  // TODO: данная строка кода позволяет загрузить данные в таблицу "andrei.Клиент". При необходимости она может быть перемещена или удалена.  this.клиентTableAdapter.Fill(this.andrei.Клиент);  // TODO: данная строка кода позволяет загрузить данные в таблицу "andrei.Автомобиль". При необходимости она может быть перемещена или удалена.  this.автомобильTableAdapter.Fill(this.andrei.Автомобиль); |

Таблица 5 – Свойства объектов формы авторизации

| Объекты | Свойства | Значения |
| --- | --- | --- |
| Form1 | Background Image | УП07.Properties.Resources.Unexpected\_repair\_bills |
|  | Text | Авторизация |
|  | Visible | True |
| Button1 | Location | 93; 28 |
|  | Size | 115; 25 |
|  | Font | Microsoft Sans Serif; 11,25pt; style=Bold |
| Label1 | Text | Авторизация |
|  | Size | 140; 24 |
|  | Location | 248; 79 |
| Label2 | Text | Логин |
|  | Size | 69; 24 |
|  | Location | 146; 111 |
| Label3 | Text | Пароль |
|  | Size | 82; 24 |
|  | Location | 146; 142 |

Таблица 5 – Свойства объектов форм «Администратор», «Директор» и «Менеджер».

| Объекты | Свойства | Значения |
| --- | --- | --- |
| Form1 | Background Image | УП07.Properties.Resources.\_48499309\_fony\_dlia\_avtoservisa\_57 |
|  | Size | 1354; 788 |
|  | Text | Администратор |
| Button1 | Location | 204; 100 |
|  | Size | 182; 39 |
|  | Font | Microsoft Sans Serif; 12pt; style=Bold |
| Button2 | Location | 204; 155 |
|  | Size | 182; 44 |
|  | Font | Microsoft Sans Serif; 12pt; style=Bold |
| Button3 | Location | 3; 290 |
|  | Size | 75; 23 |
|  | Font | Microsoft Sans Serif; 8,25pt |
| Button4 | Location | 3; 290 |
|  | Size | 75; 23 |
|  | Font | Microsoft Sans Serif; 8,25pt |
| Button5 | Location | 3; 254 |
|  | Size | 75; 23 |
|  | Font | Microsoft Sans Serif; 8,25pt |
| Button16 | Location | 464; 125 |
|  | Size | 75; 23 |
|  | Font | Microsoft Sans Serif; 8,25pt |
| Button6 | Location | 0; 100 |
|  | Size | 182; 39 |
|  | Font | Microsoft Sans Serif; 12pt; style=Bold |
| Button7 | Location | 3; 348 |
|  | Size | 75; 23 |
|  | Font | Microsoft Sans Serif; 8,25pt |
| Button8 | Location | 3; 348 |
|  | Size | 75; 23 |
|  | Font | Microsoft Sans Serif; 8,25pt |
| Button9 | Location | 3; 318 |
|  | Size | 75; 23 |
|  | Font | Microsoft Sans Serif; 8,25pt |
| Button10 | Location | 3; 319 |
|  | Size | 75; 23 |
|  | Font | Microsoft Sans Serif; 8,25pt |
| Button11 | Location | 3; 319 |
|  | Size | 75; 23 |
|  | Font | Microsoft Sans Serif; 8,25pt |
| Button12 | Location | 3; 286 |
|  | Size | 75; 23 |
|  | Font | Microsoft Sans Serif; 8,25pt |
| Button13 | Location | 154; 213 |
|  | Size | 95; 32 |
|  | Font | Microsoft Sans Serif; 9,75pt; style=Bold |
| Label1 | Text | Дата Сделки |
|  | Size | 81; 58 |
|  | Location | 133; 24 |
| Labe13 | Text | Сделка |
|  | Size | 104; 29 |
|  | Location | 169; 11 |
| Label3 | Text | Пароль |
|  | Size | 82; 24 |
|  | Location | 146; 142 |
| Panel2 | Background image | УП07.Properties.Resources.Kak\_otprazdnovat\_Vsemirnyj\_den\_rukopozhatij\_\_2048x1366 |
|  | Size | 417; 248 |
|  | Visible | True |

## 2.4 Защита данных в хранилищах

### 2.4.1 Определение привилегий пользователей с различными ролями.

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

Созданы трёх имен входа на SQL Server: Менеджер, Директор, Администратор.

Код: CREATE LOGIN Директор WITH PASSWORD = '123456'

CREATE LOGIN Менеджер WITH PASSWORD = '12345'

CREATE LOGIN Администратор WITH PASSWORD = '1234'

### 2.4.2 Настройка прав доступа к базе данных.

Используя SQL-команды в базе данных пользователям, были представлены, следующие права доступа:

Менеджер - представлен на рисунке 17. Имеет полный доступ ко всем таблицам;

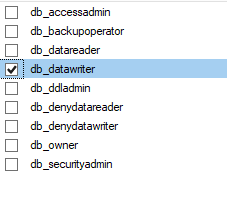


Рисунок 17 - Менеджер

Директор – Представлен на рисунке 18. Имеет возможность просмотра основных таблиц, без возможности их редактирования и чтения других таблиц;

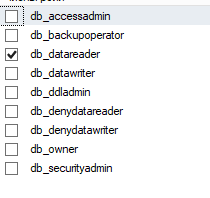


Рисунок 18 - Директор

Администратор – Представлен на рисунке 19 и имеет все права доступа базы данных.

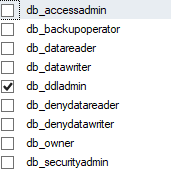


Рисунок 19 - Администратор

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

Создание резервной копии базы данных представлено на рисунке 20

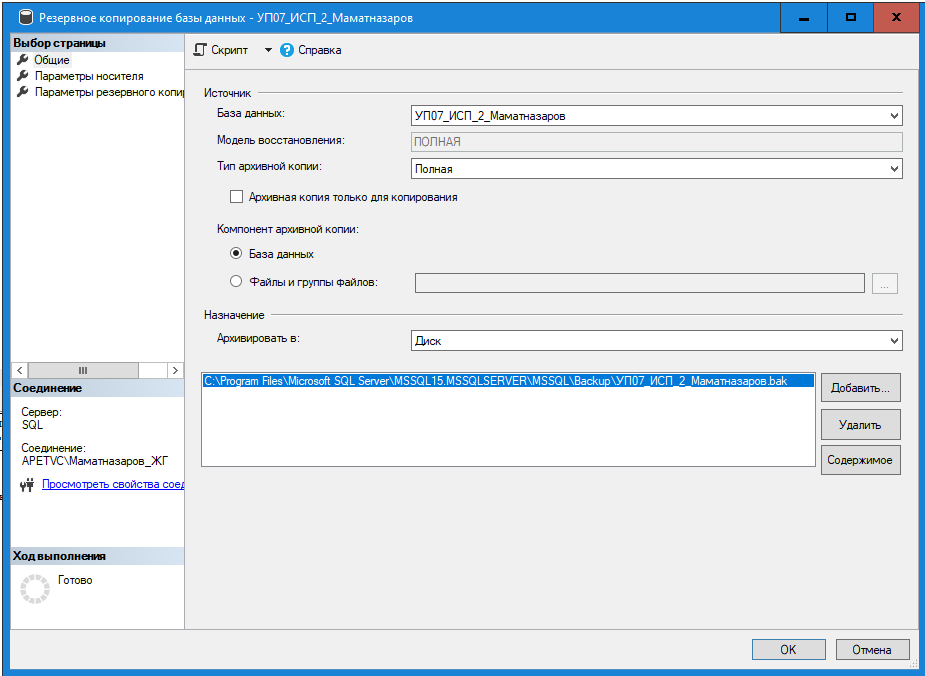


Рисунок 20 – Создание резервной копии базы данных

Восстановление резервной копии базы данных представлено на рисунке 21

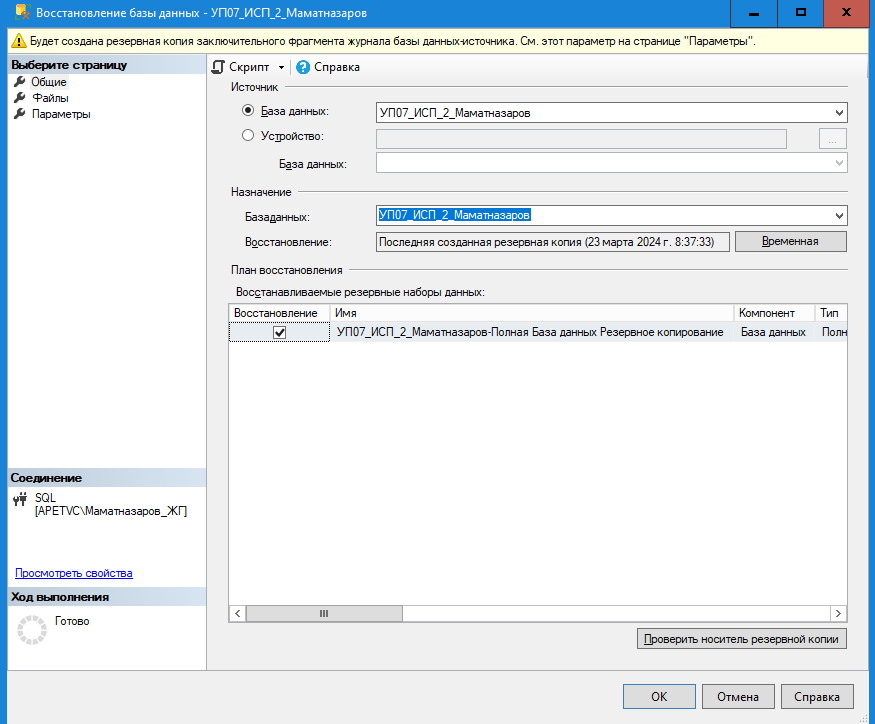


Рисунок 21 - Восстановление резервной копии базы данных

# ЗАКЛЮЧЕНИЕ

В результате практической подготовка в виде учебной практики – формы организации учебной деятельности в условиях выполнения определенных видов работ, связанных с будущей профессиональной деятельностью.

Была достигнута цель учебной практики, сформированы, закреплены и развиты практические навыки и компетенци по профилю .

Выполнены основные задачи практической подготовки в виде учебной практики, Проанализирована предметная область, а так же рассмотрены различные федеральные законы. Реализована проектная часть ИС, путём построения диаграммы вариантов использования и логической модели данных. Далее разработано БД-приложения «Техцентра» со своей серверной и клиентской частью, защитой и разграниченными правами доступа, с возможностью удаления, просмотра и добавления записей в хранилище данных из приложения.

# ПРИЛОЖЕНИЕ А

Создание физической модели базы данных

Информационная система работы с клиентами тех-центра

Ангарск 2024

Листинг программного кода

Создание базы данных

CREATE DATABASE УП07\_ИСП\_2\_Маматназаров

ON PRIMARY (

NAME= УП07\_ИСП\_2\_Маматназаров,

FILENAME='\\sql\c$\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\УП07\_ИСП\_2\_Маматназаров.mdf',

SIZE=100MB,

MAXSIZE=200, FILEGROWTH=20

)

Создание таблицы «Сотрудники»

CREATE TABLE Сотрудники

(Код\_Сотрудника INT IDENTITY(1,1)

CONSTRAINT PK1 PRIMARY KEY (Код\_Сотрудника),

Фамилия VARCHAR (50) NOT NULL,

Имя VARCHAR (50) NOT NULL,

Отчество VARCHAR (50),

Телефон CHAR (11) NOT NULL

)

Создание таблицы «Клиент»

CREATE TABLE Клиент

(Код\_Клиента INT IDENTITY(1,1)

CONSTRAINT PK3 PRIMARY KEY (Код\_Клиента),

Фамилия VARCHAR (50) NOT NULL,

Имя VARCHAR (50) NOT NULL,

Отчество VARCHAR (50),

Телефон CHAR (11) NOT NULL

)

Создание таблицы «Услуга»

CREATE TABLE Услуга

(Код\_Услуги INT IDENTITY(1,1)

CONSTRAINT PK2 PRIMARY KEY (Код\_Услуги),

Название VARCHAR (50) NOT NULL,

Цена MONEY NOT NULL,

Описание VARCHAR (200) NOT NULL

)

Создание таблицы «Автомобиль»

CREATE TABLE Автомобиль

(Код\_Автомобиля INT IDENTITY(1,1)

CONSTRAINT PK4 PRIMARY KEY (Код\_Автомобиля),

Марка VARCHAR (50) NOT NULL,

Серийный\_Номер VARCHAR (50) NOT NULL,

Описание\_Проблемы VARCHAR (400)

)

Создание таблицы «Сделка»

CREATE TABLE Сделка

(Код\_Сделки INT IDENTITY(1,1)

CONSTRAINT PK4 PRIMARY KEY (Код\_Сделки),

Код\_Услуги INT not null

CONSTRAINT FK1

FOREIGN KEY (Код\_Услуги)

REFERENCES Услуга (Код\_Услуги),

Код\_Клиента INT not null

CONSTRAINT FK2

FOREIGN KEY (Код\_Клиента)

REFERENCES Клиент (Код\_Клиента),

Код\_Сотрудника INT not null

CONSTRAINT FK3

FOREIGN KEY (Код\_Сотрудника)

REFERENCES Сотрудники (Код\_Сотрудника),

Дата\_Сделки DATE

)

Заполнение таблицы «Клиент»

INSERT INTO Клиент

VALUES ('Антонов', 'Антон','Антонович','899999999'),

('Рудской', 'Валерий','Сергеев','899999999'),

('Ленский', 'Евгений','Александрович','899999999'),

('Ленский', 'Евгений','Александрович','899999999'),

('Карапузина', 'Настасья','Петровна','899999919')

Заполнение таблицы «Сотрудники»

INSERT INTO Сотрудники

VALUES ('Заборин', 'Валентин','Андреевич','899999999'),

('Радишевский', 'Валерий','Александрович','899999999'),

('Каробкин', 'Евгений','Владимирович','899999999'),

('Нагорный', 'Виктор','Сергеевич','899999999'),

('Кондратьев', 'Георгий','Валентинович','899999919')

Заполнение таблицы «Услуга»

INSERT INTO Услуга

VALUES ('Ремонт кузова', '14000','Замена бампера'),

('Ремонт подвески', '10000','Замена бампера'),

('Ремонт двигателя', '12000','Замена поршней'),

('Замена каробки передач', '20000','Покупка и замена коробки передач'),

('Замена двигателя', '30000','Покупка и замена двигателя')

Заполнение таблицы «Автомобиль»

INSERT INTO Автомобиль

VALUES ('BMW', '12A14D4','Движок'),

('MERSEDES', 'A234JJ','Подвеска'),

('SKODA', '7880990AD','Трансмиссия'),

('TOYOTA', '121G41412','Кузов, левое крыло'),

('NISSAN', '121314HB','Капот')

Заполнение таблицы «Сделка»

INSERT INTO Сделка

VALUES (1,2,1,1,'2004-12-20'),

(1,2,2,2,'2004-12-20'),

(1,2,3,3,'2004-12-20'),

(1,2,4,4,'2004-12-20'),

(1,2,5,5,'2004-12-20'),

(2,1,5,1,'2004-12-20'),

(2,2,4,2,'2004-12-20'),

(2,3,3,4,'2004-12-20'),

(2,4,2,3,'2004-12-20'),

(2,5,1,5,'2004-12-20'),

(3,1,5,1,'2004-12-20'),

(3,2,4,2,'2004-12-20'),

(3,3,3,5,'2004-12-20'),

(3,4,2,4,'2004-12-20'),

(3,5,1,3,'2004-12-20')

# 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.