# ВВЕДЕНИЕ

Тема выпускной квалификационной работы – «Разработка информационной системы по сопровождению продаж инструментов красоты».

В настоящее время ведение полноценного современного бизнеса невозможно представить без помощи Интернет-ресурсов. Информационные системы предоставляют клиентам возможность с помощью сети Интернет просматривать ассортимент продукции, изучать характеристики товаров. Эффективные продажи в сети Интернет на данный момент осуществляются с помощью интернет-магазинов. Люди всё больше предпочитают осуществлять покупки именно через интернет.

С помощью интернет-магазинов существует возможность выполнить заказ в любое время суток, а также выбрать удобное место и время доставки товара покупателю. Кроме того интернет-магазин более удобен по сравнению с обычным магазином возможностью просмотра всего ассортимента товара с настройкой необходимых фильтров и поиском нужного товара.

В условиях пандемии короновируса информационные системы дистанционного взаимодействия крайне актуальны, в том числе и в сфере индустрии красоты.

Услуги парикмахера и мастера ногтевого сервиса на сегодняшний день пользуются большим спросом. Индустрия красоты развивается и растет с каждым днем. Каждый человек, который ухаживает за собой, хотя бы один раз в год посещает салон красоты. Большое внимание в этой сфере уделяется качеству услуг. И главное значение в этом играют инструменты, которыми пользуются мастера. Качественный инструмент – это одно из важнейших условий успешного выполнения услуги в сфере красоты.

Для упрощения поиска интересующих покупателя инструментов красоты появилась необходимость в информационной системе по сопровождению продаж инструментов красоты. Таким образом, тема выпускной квалификационной работы является актуальной.

Объектом выпускной квалификационной работы являются технологии автоматизации электронной торговли и учета.

Предмет исследования – разработка информационной системы по сопровождению продаж инструментов красоты.

Целью выпускной квалификационной работы является разработка информационной системы по сопровождению продаж инструментов красоты.

Для достижения поставленной цели необходимо выполнить следующие задачи:

* изучить предметную область;
* проанализировать структуру будущего приложения;
* выделить требования к системе;
* разработать настольное приложение на платформе .NET;
* оформить документацию.

Подборка необходимых материалов осуществляется с использованием сети интернет и соответствующей литературы по разработке приложений.

Работа состоит из пояснительной записки, состоящей из \_\_ страниц текста, содержащей \_\_ рисунков, \_\_ таблиц, библиографический список из 18 источников. Структура работы соответствует логике исследования, и включает в себя введение, основную часть, специальную часть, стратегии продвижения, тестирование, заключение, список литературы, \_\_ приложений.

1 ОБЩАЯ ЧАСТЬ

* 1. Описание предметной области

Деятельность магазина связана с продажей товаров красоты. В магазин поставляется следующие инструменты:

* Косметика для волос: шампуни, бальзамы, кондиционеры, макси для волос;
* Парикмахерские инструменты: машинки для стрижки волос, триммеры для стрижки волос, фены для волос;
* Для ногтей: гель лаки, базы, топы, жидкости для маникюра;
* Расходные материалы: дезинфекция, перчатки, макси, полотенца.

Покупатель, находясь в магазине, может выбрать нужные ему товар и пройти на кассу, при покупке товара покупатель может выбрать между наличной и безналичной оплатой, после совершения оплаты, происходит списание товара со склада.

В магазине работают следующие сотрудники:

- Администратор – ведет в Excel-е товарооборот;

- Кладовщик сверяет груз с накладной и делает сверку: наименование товаров, их количество и цены. Для этого вскрывают все транспортные упаковки или несколько на выбор, если к поставщику есть доверие.

Если товара не хватает, обычно из накладной вычеркивают строки с товарами, которые забыли поставить. У каждой пометки ответственный сотрудник ставит подпись. Потом документ заменяют на верный экземпляр.

После приемки от поставщика товар отправляется на склад магазина, на каждую поставку от контрагента составляется накладная в бумажном виде, и в дальнейшем работники раскладывают товар по полкам в магазине в соответствии с их категорией.

Введение товарооборота в Excel-е затрудняет процесс обработки информации, поэтому я сделаю информационную систему по сопровождению продаж инструментов красоты чтобы сделать работу магазина эффективнее.

* 1. Сравнение аналогов

-

Таблица 1 – Сравнительные характеристики аналогов интернет-магазина

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Функционал |  |  |  |  |  |
| регистрационная карточка | – | – | – | – | – |
| происк | – | – | – | – | – |
| информация для пользователя | + | + | + | + | + |
| отдельная окно с товарами | – | – | + | – | + |
| возможность оставить отзывы | + | + | + | + | + |
| прописанные  сроки доставки | + | + | + | + | + |
| контактные данные | + | + | + | + | + |
| описания товара | + | + | + | + | + |
| различные способы оплаты | + | + | + | + | + |
| разделение товара по категориям | + | – | + | – | + |
| сортировка товара | + | – | + | – | + |
| связь с менеджером | + | + | + | + | + |

1.3 Формирование функциональных требований

Необходимо разработать информационную систему, которая сможет решить проблему покупки инструментов красоты.

На сайте должен быть представлен каталог с товарами. Каждый товар должен иметь свою категорию и информацию:

* наименование;
* стоимость;
* описание;
* изображение;
* количество на складе;
* категорию.

Пользователь сможет выполнить поиск и сортировку необходимых ему инструментов, после покупки система должна добавить пользователя в базу данных с информацией:

* ФИО;
* телефон;
* email;
* адрес.

Клиент может сразу заказать нужный ему инструмент, а также может добавить понравившийся товар в корзину и продолжить искать инструменты в каталоге. После оплаты, заказ должен быть записан в базу данных с информацией:

* клиент;
* товар;
* дата покупки.

При авторизации в приложении пользователь должен получить роль с определенным набором прав:

* пользователь – должен иметь доступ просмотру каталога и контактной информации;
* директор – должен иметь доступ к правке категорий, комплектующих, клиентов и иметь доступ к просмотру отчетности.

## Проектирование поведения системы

Диаграмма деятельности представляет собой блок–схему, которая наглядно показывает, как поток управления переходит от одной деятельности к другой.

Для информационной системы были выделены следующие действующие потоки: система и клиент.

Были выделены следующие действия:

* выбор товара;
* ввод контактных данных;
* вывод каталога товаров;
* выставление платежного счета;
* добавление записи продажи в базу данных;
* отправка сообщения о покупке пользователю на почту.

Каждому действующих потоков были назначены соответствующие действия:

* Действующей поток «Клиент» связан с действием «Выбор товара» и «Ввод контактных данных».
* Действующей поток «Система» связан с действием «Вывести каталог товаров», «Выставить платежный счет», «Добавление записи продажи в базу данных», «Отправка сообщения о покупке пользователю на почту».

На проектировании показано назначение действующим потоков действий в виде диаграммы «Activitydiagram» с процессом записи клиента на услугу(смотри приложение Б).

## Проектирование физического представления

Диаграммы развертывания используются для визуализации аппаратных процессоров/узлов/устройств системы, каналов связи между ними и размещения программных файлов на этом аппаратном обеспечении.

Для информационной системы были выделены следующие действующие ноды: Browser, DesktopClient и WebServer.

1. DesktopClient содержит в себе модуль «BeautyStore.exe»;
2. WebServer содержит в себе артефакт «APS.NET 4x», «MSSQLServer».
3. Browser содержит в себе артефакт «Browser» и модуль «BeautyStoreWebApplication».

На рисунке 7 представлено физическое проектирование информационной системы в виде диаграммы «Deploymentdiagram».

(смотри Приложение В).

## 1.5 Проектирование базы данных

### 1.5.1 Семантическая модель

Семантическая модель – модель предметной области, предназначенная для представления семантики предметной области на самом высоком уровне абстракции.

Для информационной системы были выделены следующие действующие сущности:

* категории;
* товары;
* клиенты.

В результате проектировании показана семантическая модель информационной системы(смотри Приложение Г).

### 1.5.2 Логическая модель

Логическая модель базы данных – схема базы данных, выраженная в понятиях модели данных. Этим отличается от концептуальной модели, описывающей семантику предметной области без указания технологии (конкретных методов реализации), и от физической модели, которая описывает конкретные физические механизмы, применяемые для хранения данных в накопителях.

Для информационной системы были выделены следующие сущности:

* администраторы;
* права доступа;
* клиенты;
* товары;
* категории;
* покупки.

Сущность «Администраторы» имеет следующие поля:

* уникальный идентификатор;
* права доступа;
* логин;
* пароль.

Сущность «Права доступа» имеет следующие поля:

* уникальный идентификатор;
* доступ к клиентам;
* доступ к товарам;
* доступ к продажам.

Сущность «Товары» имеет следующие поля:

* уникальный идентификатор;
* категория;
* наименование;
* стоимость;
* изображение;
* количество на складе.

Сущность «Категории» имеет следующие поля:

* уникальный идентификатор;
* наименование.

Сущность «Продажи» имеет следующие поля:

* уникальный идентификатор;
* клиент;
* товар;
* дата продажи.

В результате проектирования показана логическая модель информационной системы (смотри Приложение Д).

## 1.6 Проектирование интерфейса

Интерфейс – это вся видимая пользователю часть сервиса, с которой он взаимодействует, решая свои задачи.

В результате проектирования показано проектирование интерфейса для информационной системы (смотри Приложение Е).

# 2 СПЕЦИАЛЬНАЯ ЧАСТЬ

## 2.1 Выбор технологий разработки

### 2.1.1 Выбор технологий разработки БД

База данных (БД) – именованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области.

Под предметной областью принято понимать некоторую область человеческой деятельности или область реального мира, подлежащих изучению для организации управления и автоматизации, например, предприятие, вуз и т.д. MySQL – свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой SunMicrosystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU GeneralPublicLicense, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL–лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.

Microsoft SQL Server – система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов – Transact–SQL, создан совместно Microsoft и Sybase. Transact–SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.

MongoDB –документно–ориентированная система управления базами данных (СУБД) с открытым исходным кодом, не требующая описания схемы таблиц. Классифицирована как NoSQL, использует JSON–подобные документы и схему базы данных. Написана на языке C++. SQLite – компактная встраиваемая СУБД. Исходный код библиотеки передан в общественное достояние. В 2005 году проект получил награду Google–O’ReillyOpenSourceAwards.

В создании приложения задействована База Данных MSSQL. Так как она идеально подходит под функции приложения и с ней удобно работать на C#.

### 2.1.2 Выбор технологий клиентской части

XAML (eXtensibleApplicationMarkupLanguage) - язык разметки, используемый для инициализации объектов в технологиях на платформе .NET. Применительно к WPF (а также к Silverlight) данный язык используется прежде всего для создания пользовательского интерфейса декларативным путем. Хотя функциональность XAML только графическими интерфейсами не ограничивается: данный язык также используется в технологиях WCF и WF, где он никак не связан с графическим интерфейсом. То есть его область шире. Применительно к WPF мы будем говорить о нем чаще всего именно как о языке разметки, который позволяет создавать декларативным путем интерфейс, наподобие HTML в веб-программировании. Однако опять же повторюсь, сводить XAML к одному интерфейсу было бы неправильно, и далее на примерах мы это увидим.

XAML - не является обязательной частью приложения, мы вобще можем обходиться без него, создавая все элементы в файле связанного с ним кода на языке C#. Однако использование XAML все-таки несет некоторые преимущества:

Возможность отделить графический интерфейс от логики приложения, благодаря чему над разными частями приложения могут относительно автономно работать разные специалисты: над интерфейсом - дизайнеры, над кодом логики - программисты.

Компактность, понятность, код на XAML относительно легко поддерживать.

При компиляции приложения в VisualStudio код в xaml-файлах также компилируется в бинарное представление кода xaml, которое называется BAML (BinaryApplicationMarkupLanguage). И затем код baml встраивается в финальную сборку приложения - exe или dll-файл.

HTML (от англ. HyperTextMarkupLanguage — «язык гипертекстовой разметки») — стандартизированный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML (или XHTML). Язык HTML интерпретируется браузерами; полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства.

Язык HTML до 5-й версии определялся как приложение SGML (стандартного обобщённого языка разметки по стандарту ISO 8879). Спецификации HTML5 формулируются в терминах DOM (объектной модели документа).

Язык XHTML является более строгим вариантом HTML, он следует синтаксису XML и является приложением языка XML в области разметки гипертекста.

Во всемирной паутине HTML-страницы, как правило, передаются браузерам от сервера по протоколам HTTP или HTTPS, в виде простого текста или с использованием шифрования.

В HTML можно встроить программый код на языке программирования JavaScript, для управления поведением и содержанием веб-страниц. Также включение CSS в HTML описывает внешний вид и макет страницы.

Данный язык идеально подойдет для моего проекта, так как вместе с ним можно использовать фреймворкBootstrap, который значительно облегчит написания логики представления и позволит более серьезно сконцентрироваться на написании бизнес логики приложения.

Для модуля «Администратор» будет использоваться технология WPF, где логика представления будет написана на XAML.

Для модуля «Клиент» будет использоваться HTML, CSS и JS так как это веб приложения на ASP.NET.

### 2.1.3 Выбор технологий серверной части

В качестве серверного языка был выбран C#, на сегодняшний момент язык программирования C# один из самых мощных, быстро развивающихся и востребованных языков в ИТ-отрасли. В настоящий момент на нем пишутся самые различные приложения: от небольших десктопных программок до крупных веб-порталов и веб-сервисов, обслуживающих ежедневно миллионы пользователей.

C# уже не молодой язык и кFк и вся платформа .NET уже прошел большой путь. Первая версия языка вышла вместе с релизом MicrosoftVisualStudio .NET в феврале 2002 года. Текущей версией языка является версия C# 9.0, которая вышла 10 ноября 2020 года вместе с релизом .NET 5.

C# является языком с Си-подобным синтаксисом и близок в этом отношении к C++ и Java. Поэтому, если вы знакомы с одним из этих языков, то овладеть C# будет легче.

C# является объектно-ориентированным и в этом плане много перенял у Java и С++. Например, C# поддерживает полиморфизм, наследование, перегрузку операторов, статическую типизацию. Объектно-ориентированный подход позволяет решить задачи по построению крупных, но в тоже время гибких, масштабируемых и расширяемых приложений. И C# продолжает активно развиваться, и с каждой новой версией появляется все больше интересных функциональностей, как, например, лямбды, динамическое связывание, асинхронные методы и т.д.

## 2.2 Разработка базы данных

Для разрабатываемой информационной системы создана таблица «Товары» с полями: «Id», «IdCategory», «Customer» «Name», «Cost», «Img», «Quantity», создание таблицы «Товары» показано на рисунке 5.

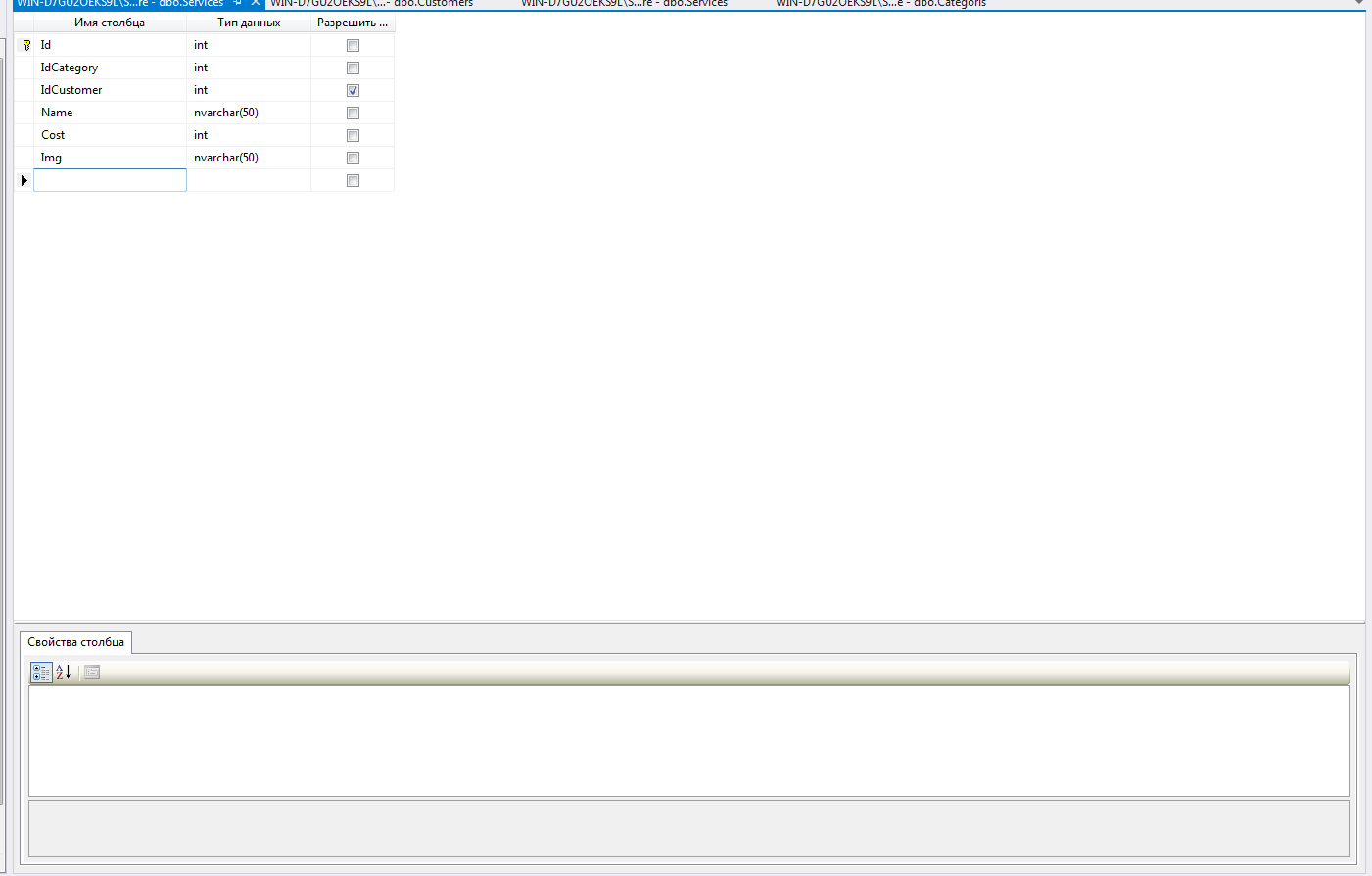


Рисунок 5 – Создание таблицы «Товары»

Создана таблица «Категории» с полями: «Id», «Name», создание таблицы «Категории» показано на рисунке 6.

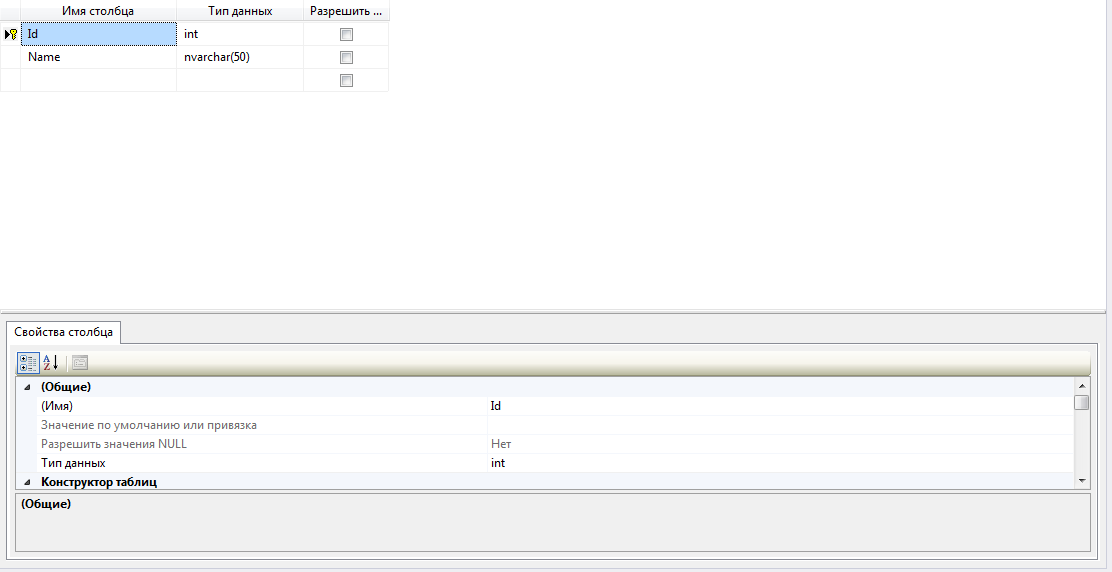


Рисунок 6 – Создание таблицы «Категории»

Создана таблица «Клиенты» с полями: «Id», «FullName», «Phone», создание таблицы «Клиенты» показано на рисунке 7.

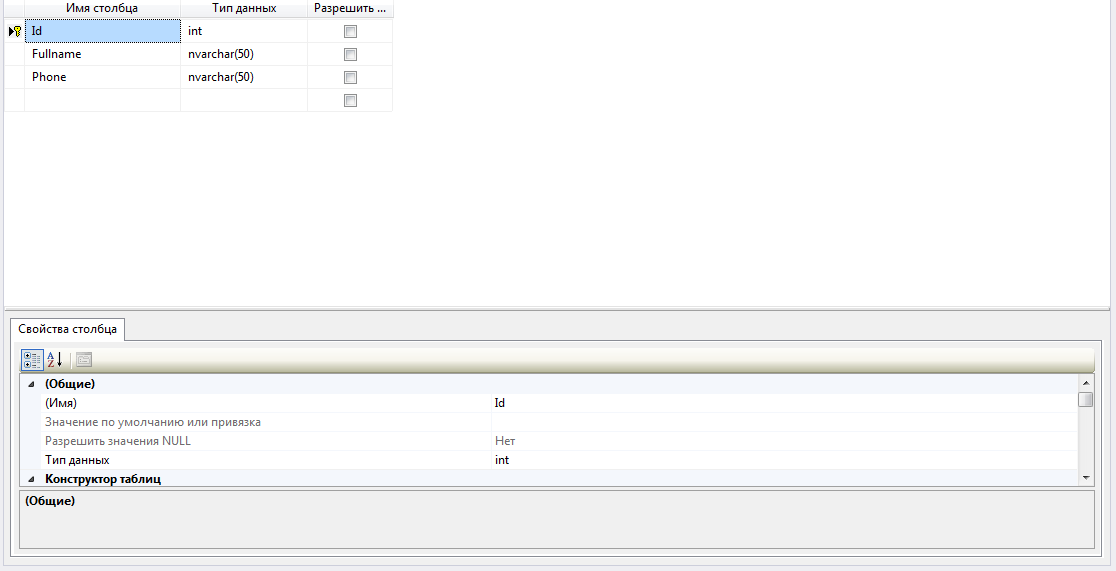


Рисунок 7 – Создание таблицы «Клиенты»

Создана таблица «Администраторы» с полями: «Id», «Login», «Password», «PermissionId», создание таблицы «Администраторы» показано на рисунке 8.

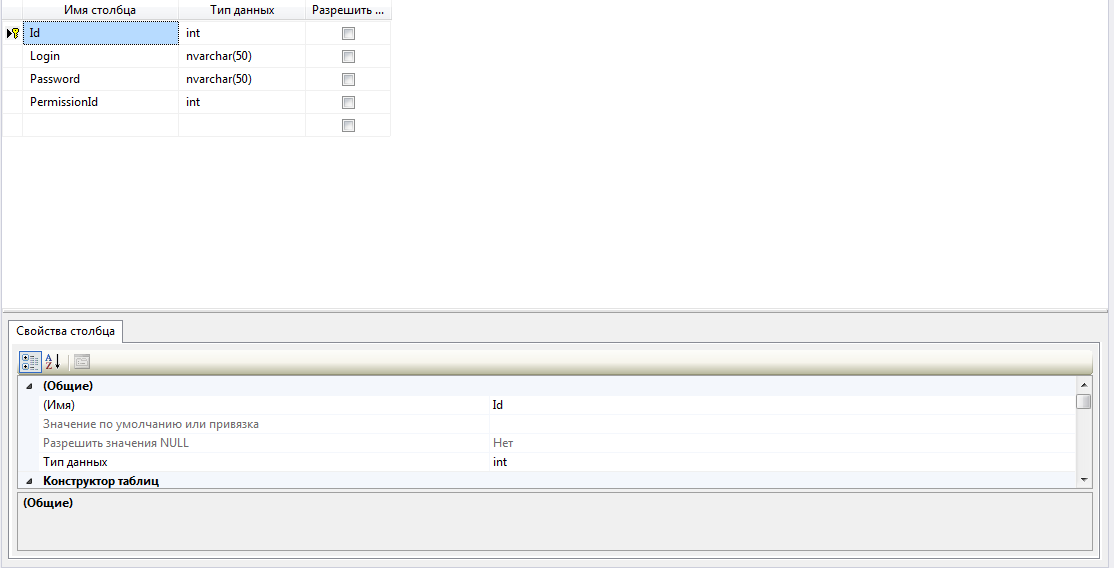


Рисунок 8 – Создание таблицы «Администраторы»

Создана таблица «Права доступа» с полями: «Id», «Name», «AccessServices», «AccessClients», «AccessClientService», «AccessReports», создание таблицы «Права доступа» показано на рисунке 9.

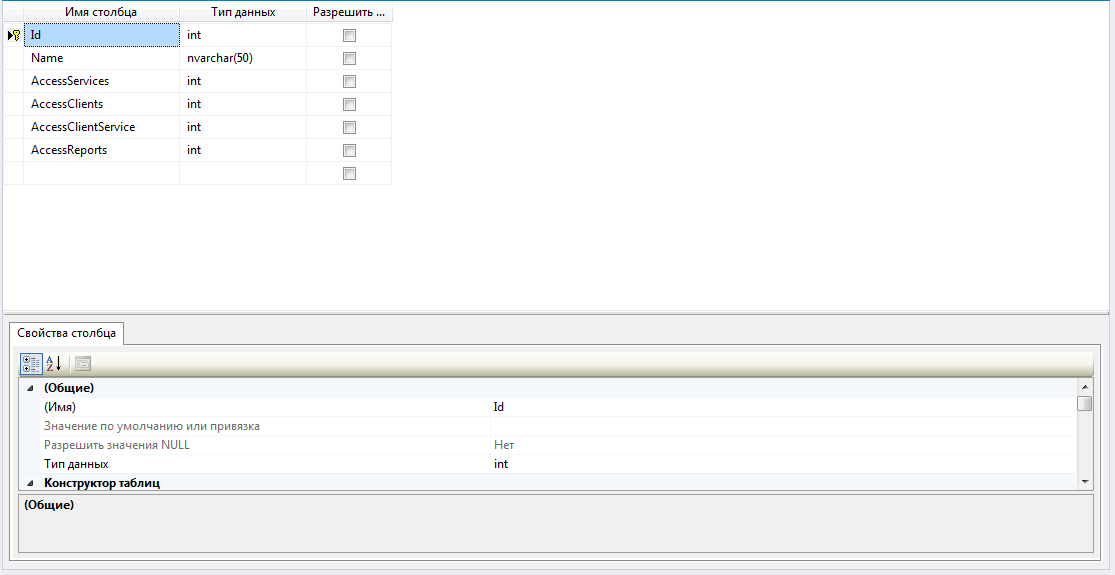


Рисунок 9 – Создание таблицы «Права доступа»

Создана таблица «Поставщики» с полями: «Id», «Name», создание таблицы «Поставщики» показано на рисунке 8.

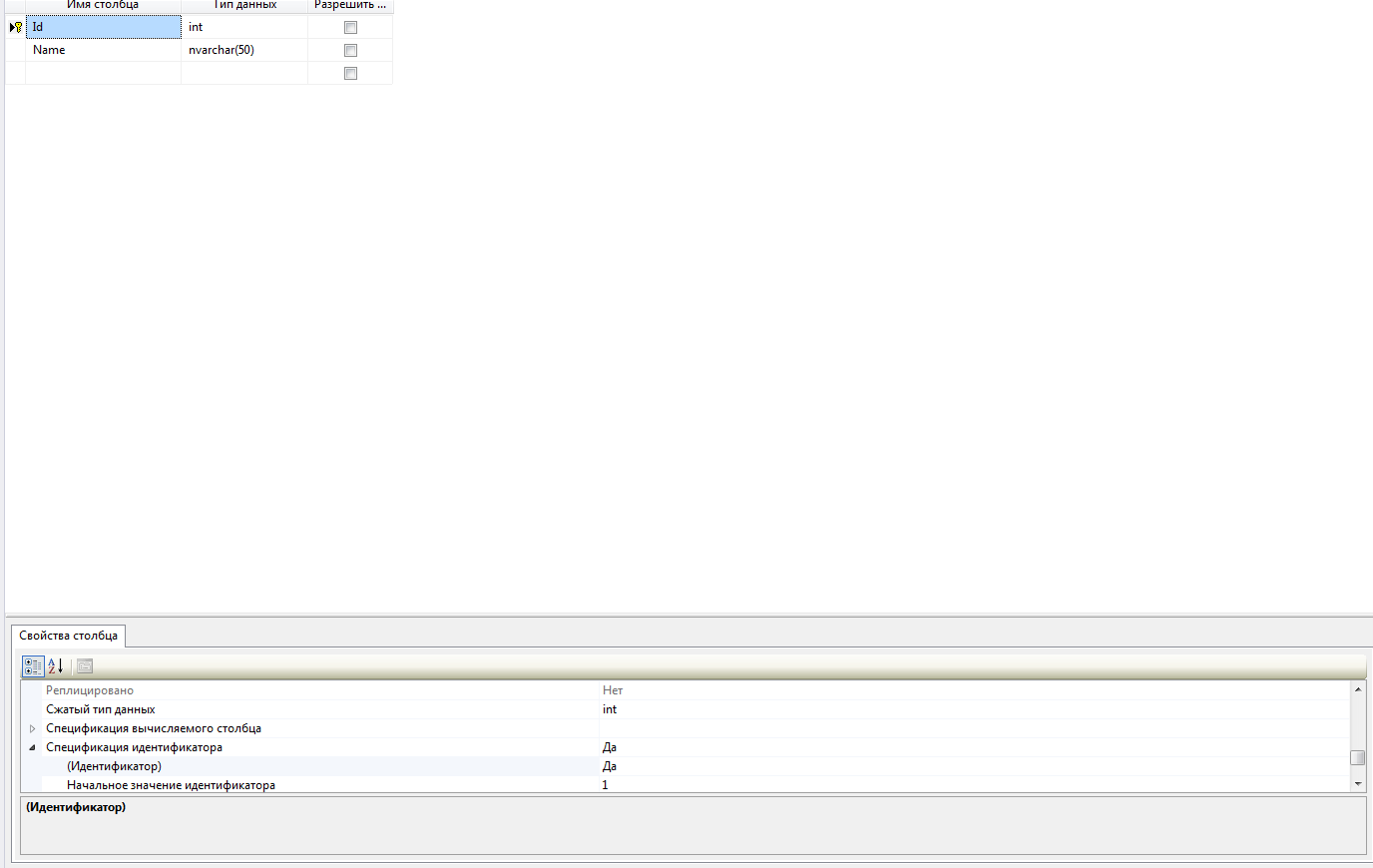


Рисунок 9 – Создание таблицы «Поставщики»

Создание связи между таблицами «Администраторы и «Права доступа» показано на рисунке 10.

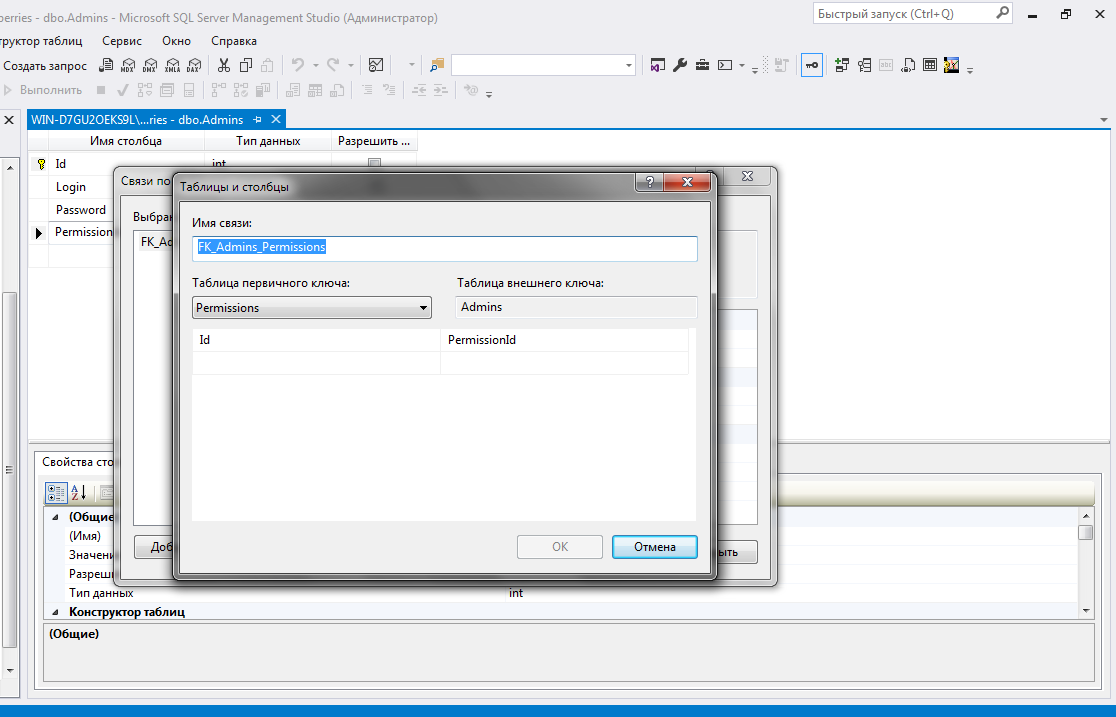


Рисунок 10 – Создание связи

Создание связи между таблицами «Товары» и «Категории» показано на рисунке 11.

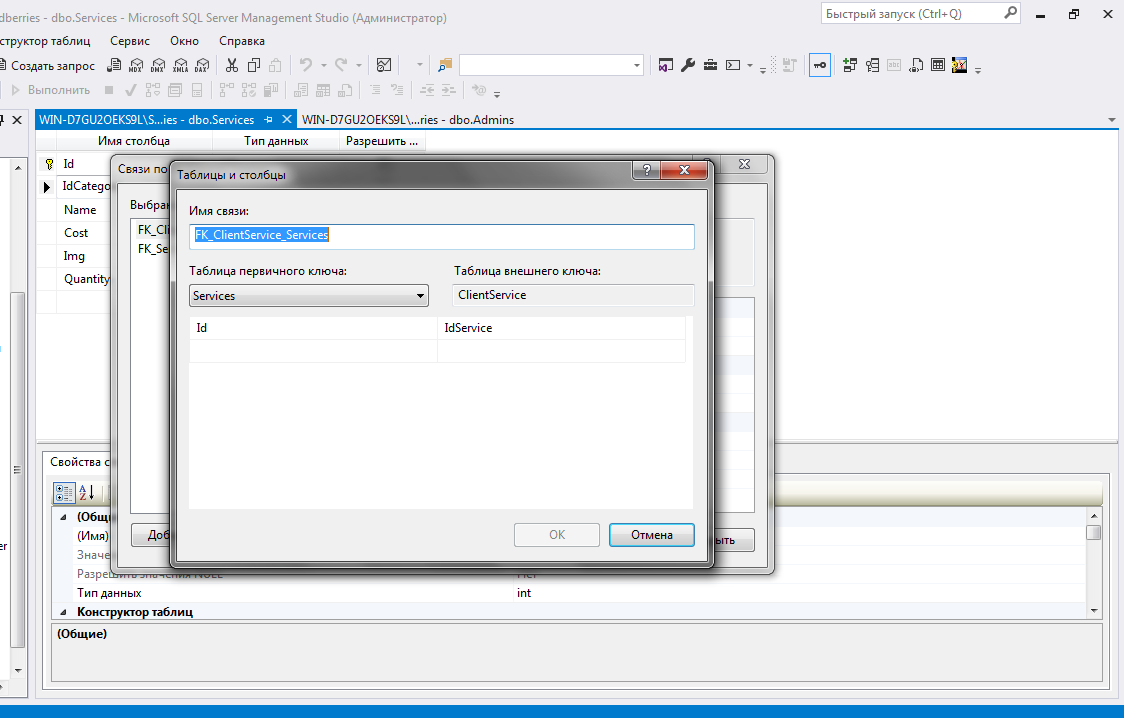


Рисунок 11 – Создание связи

Создание связи между таблицами «Клиенты» и «Покупки» показано на рисунке 12.

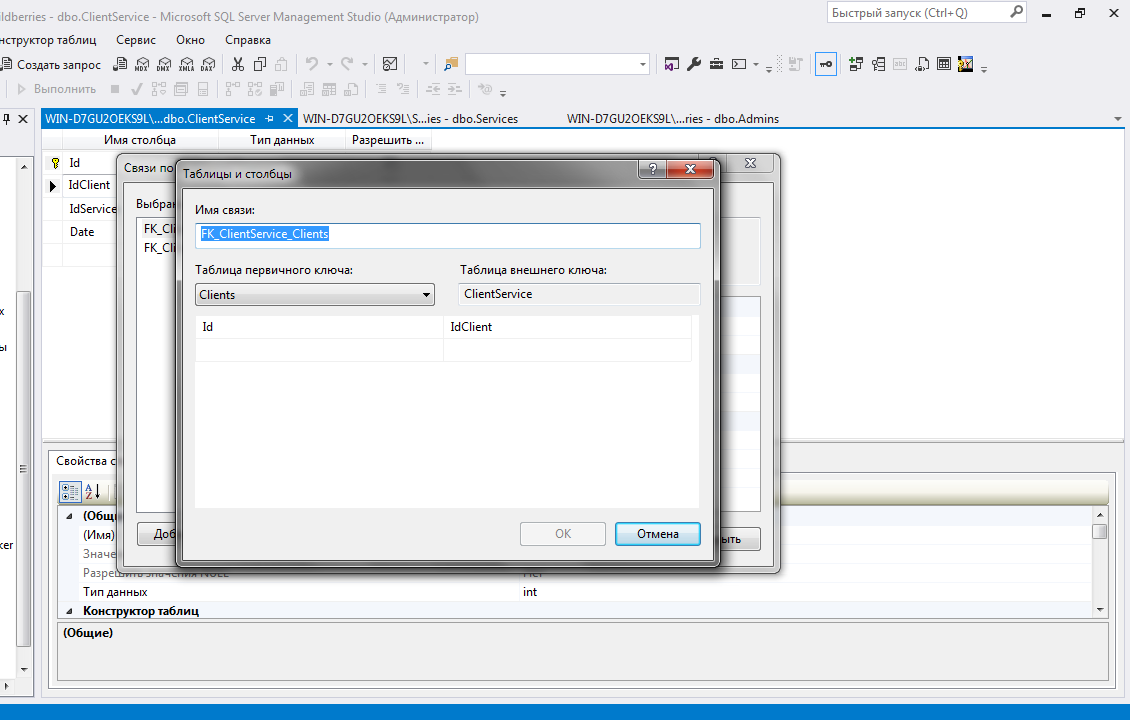


Рисунок 12 – Создание связи

Создание связи между таблицами «Товары» и «Продажи» показано на рисунке 13.

.

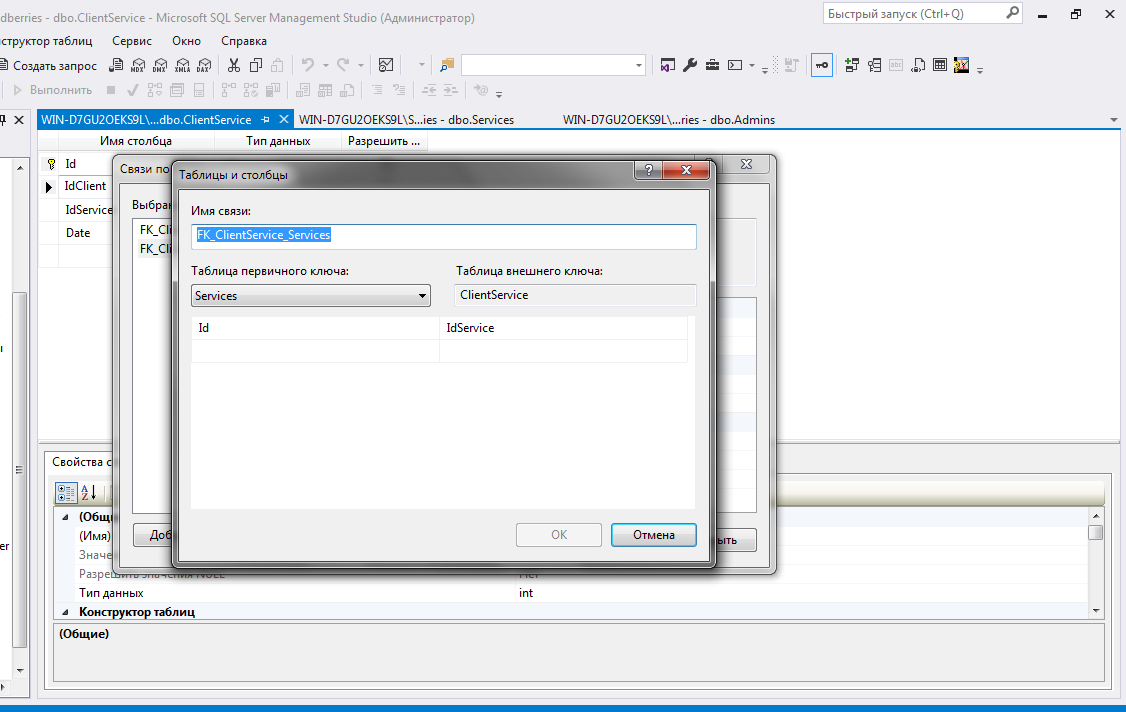


Рисунок 13 – Создание связи

Создание связи между таблицами «Товары» и «Поставщики» показано на рисунке 13.

.

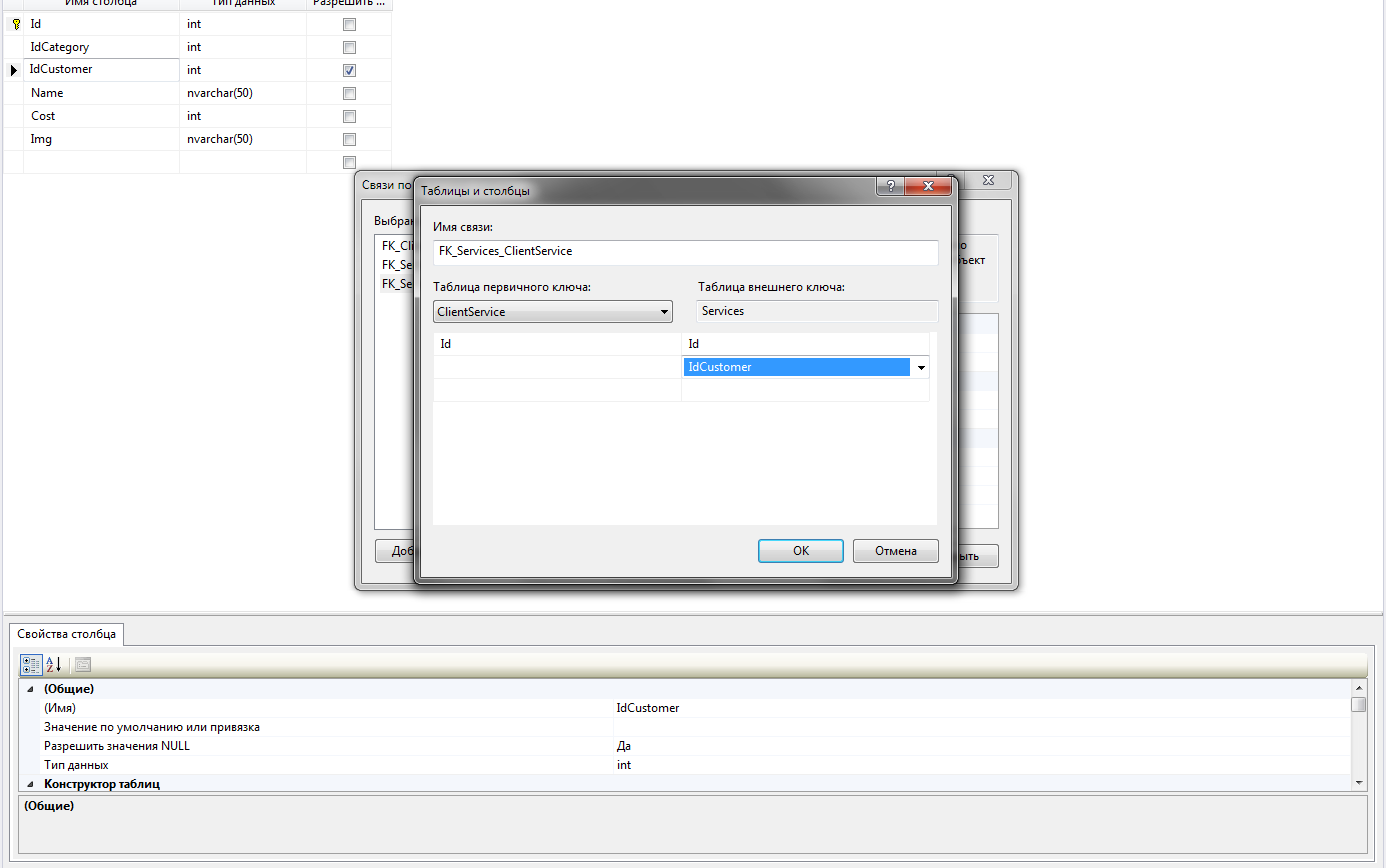


Рисунок 13 – Создание связи

Заполнение таблицы «Права доступа» тестовыми данными показано на рисунке 14.

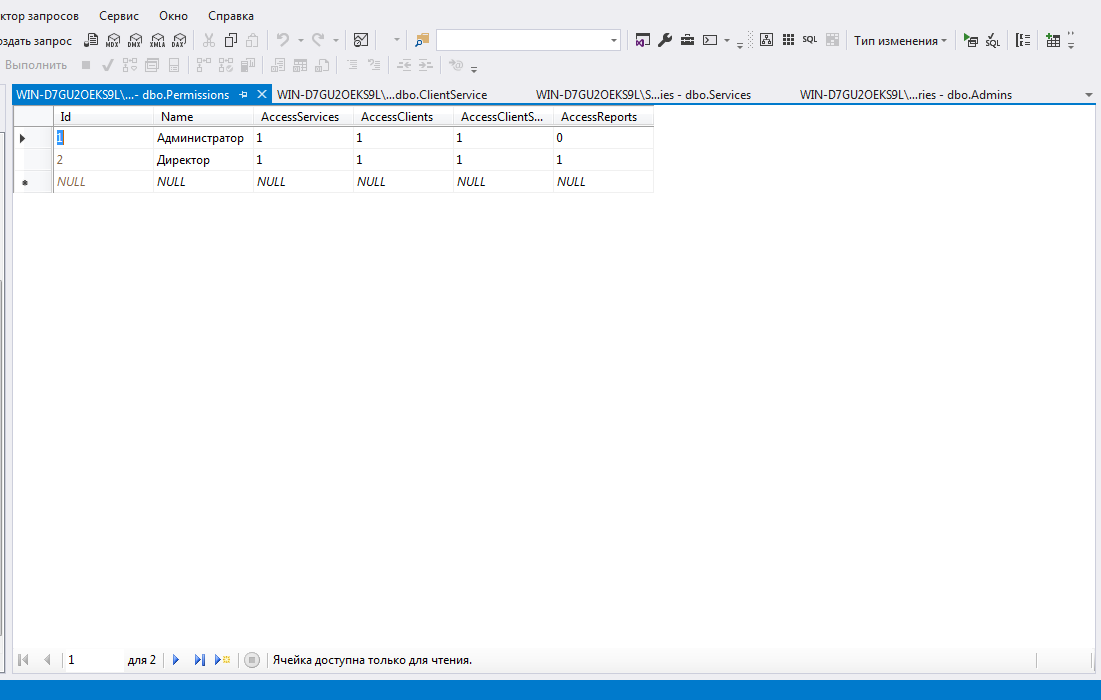


Рисунок 14 – Заполнение таблицы «Права доступа»

Заполнение таблицы «Администраторы» тестовыми данными показано на рисунке 15.

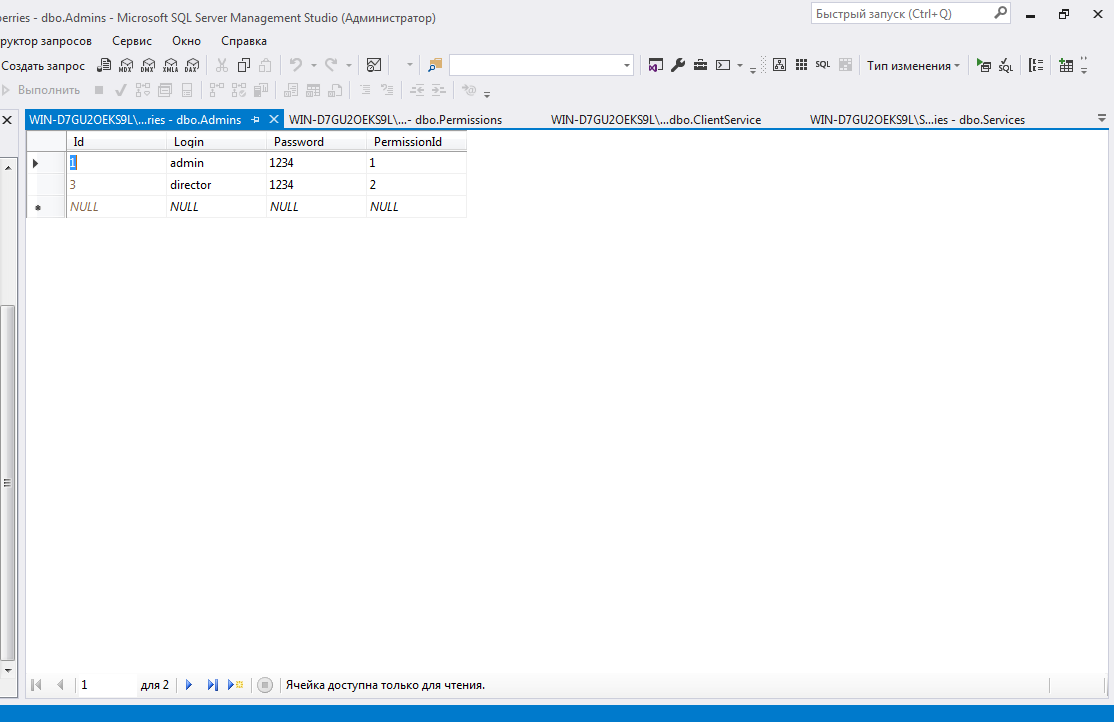


Рисунок 15 – Заполнение таблицы «Администраторы»

Заполнение таблицы «Клиенты» тестовыми данными показано на рисунке 16.

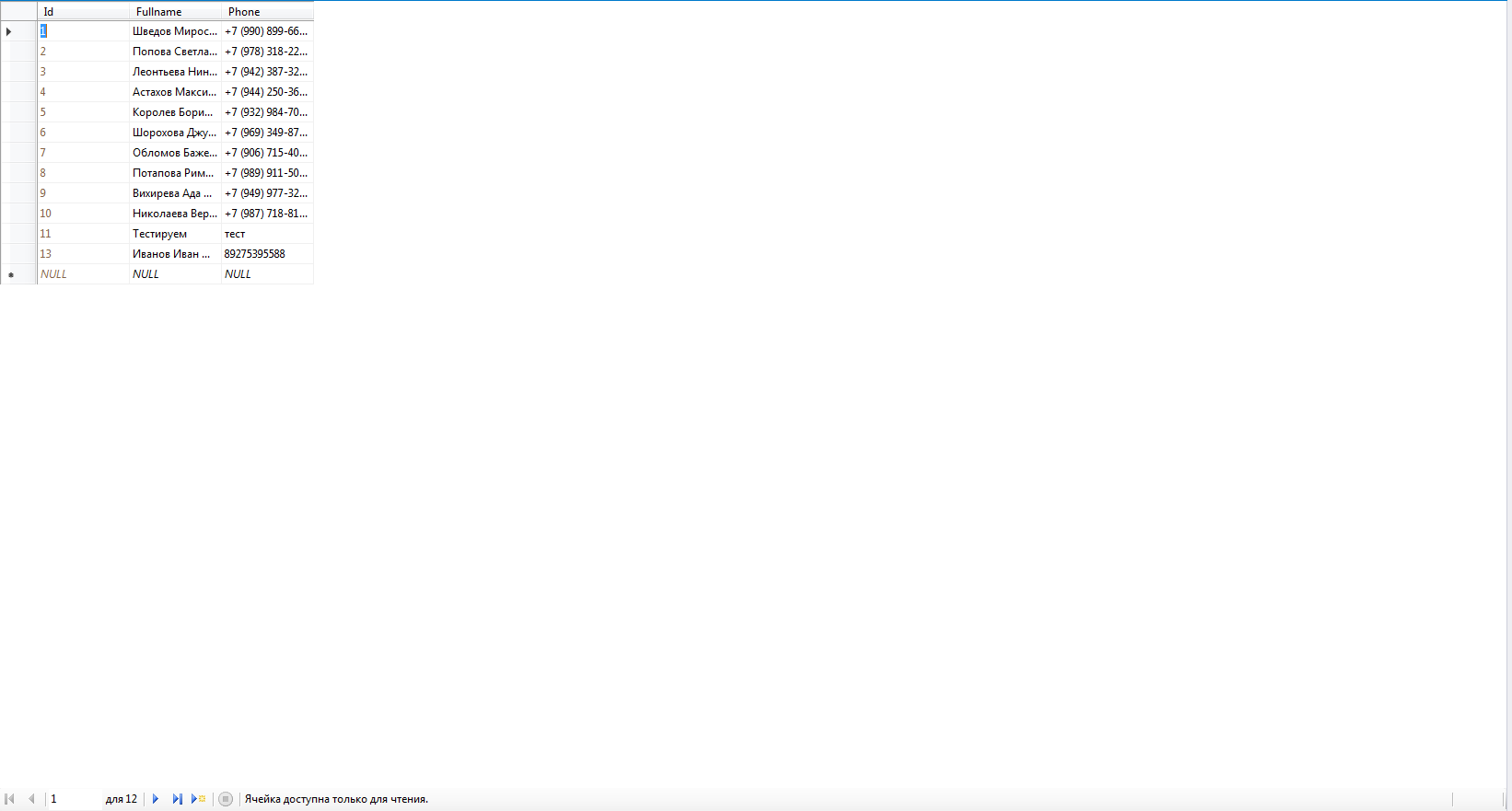


Рисунок 16 – Заполнение таблицы «Клиенты»

Заполнение таблицы «Товары» показано на рисунке 17.

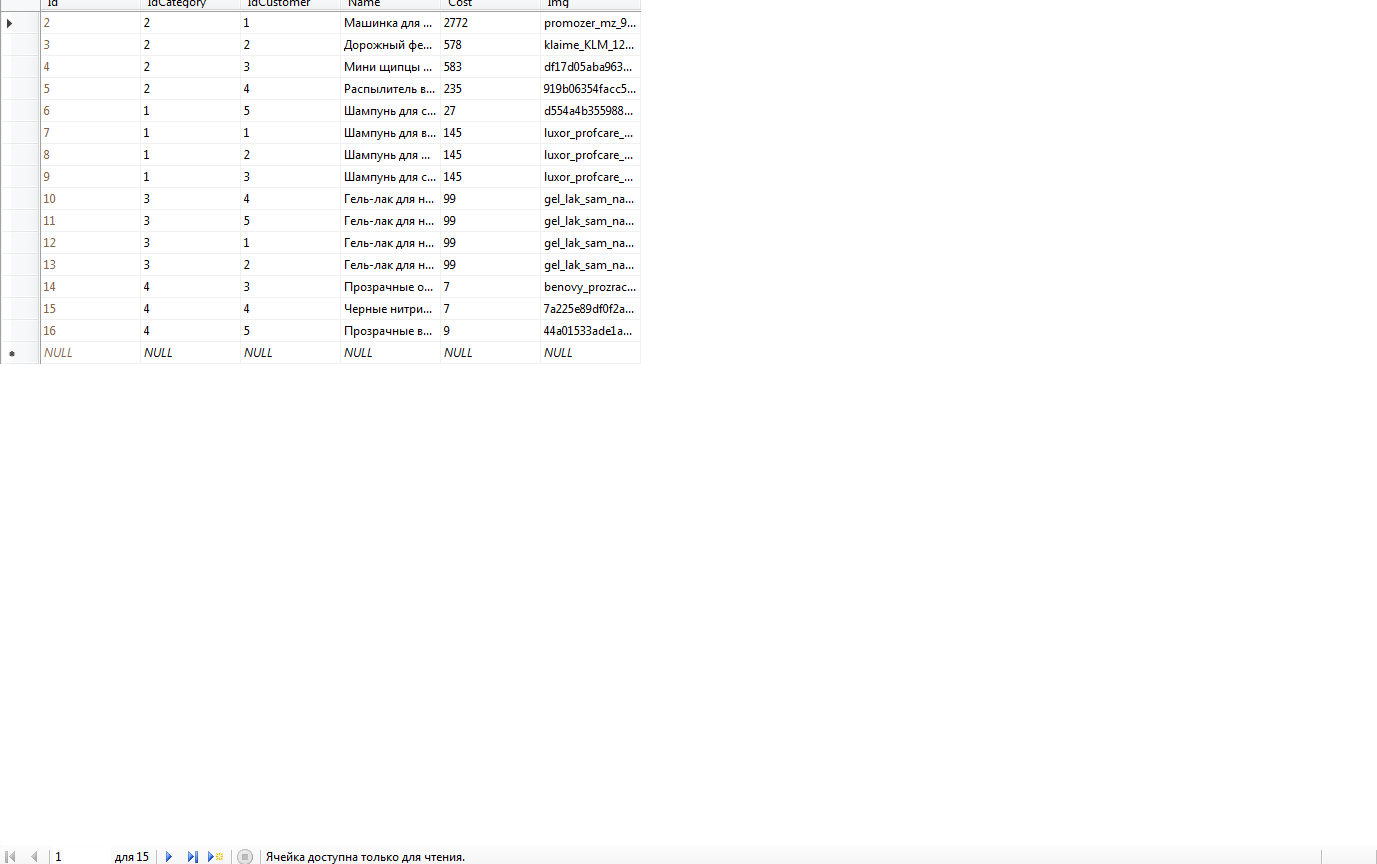


Рисунок 17 – Заполнение таблицы «Товары»

Заполнение таблицы «Категории» тестовыми данными показано на рисунке 18.

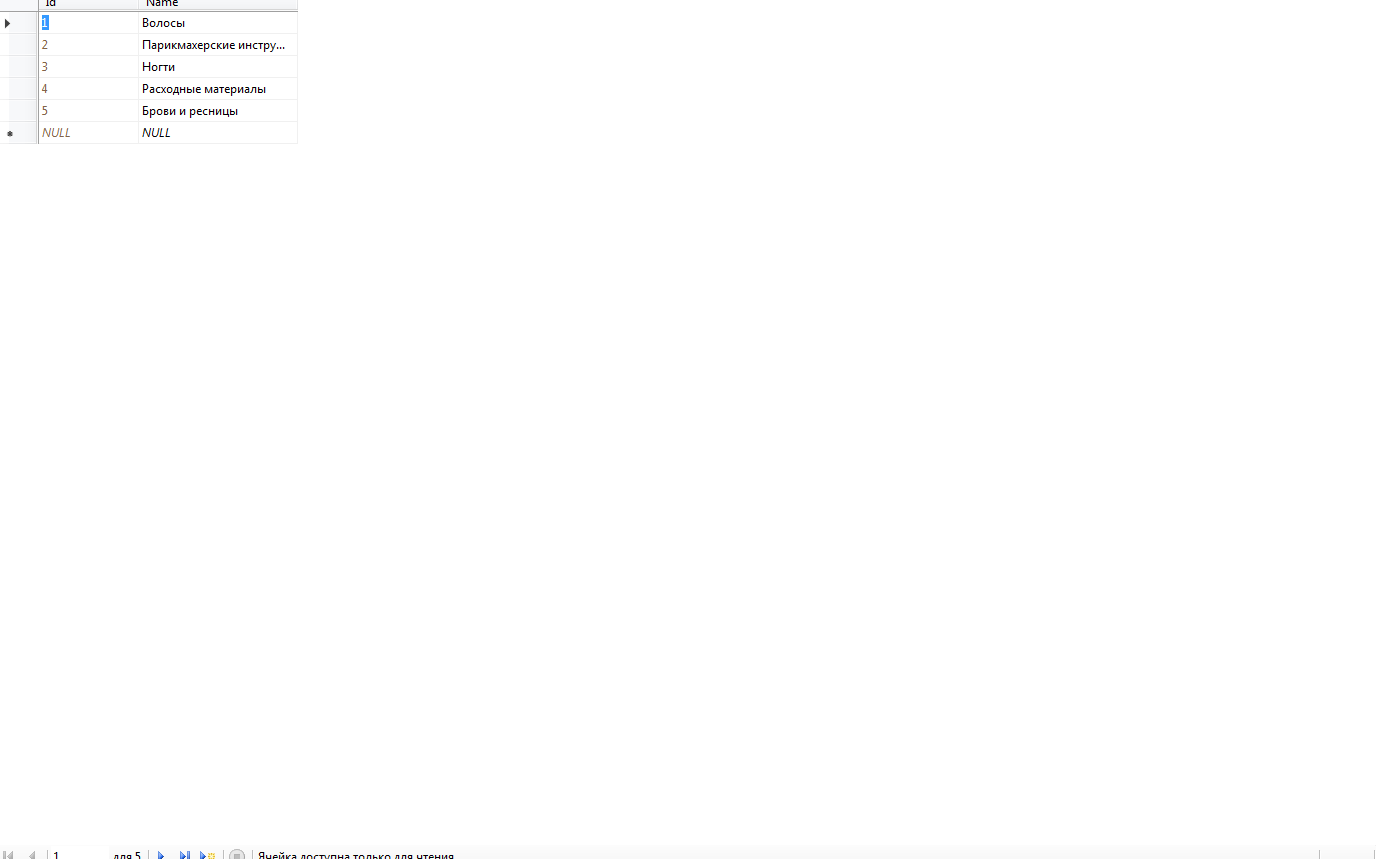


Рисунок 18 – Заполнение таблицы «Категории»

Заполнение таблицы «Продажи» тестовыми данными показано на рисунке 19.

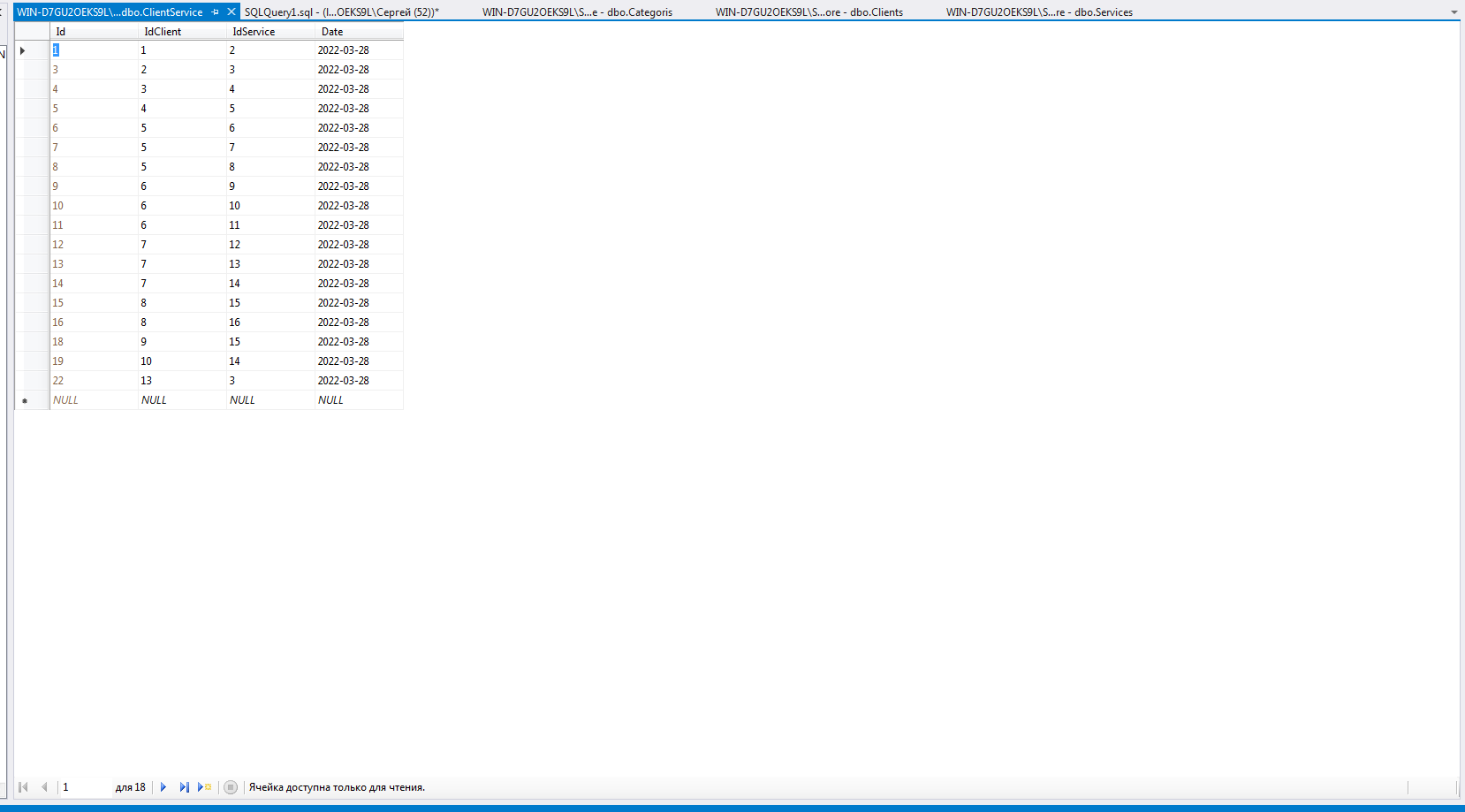


Рисунок 19 – Заполнение таблицы «Продажи»

Заполнение таблицы «Поставщики» тестовыми данными показано на рисунке 19.



Рисунок 19 – Заполнение таблицы «Поставщики»

## 2.3 Разработка клиентского приложения

Создание страницы вывода, редактирование и удаления товаров показано на рисунке 31. На данной странице администратор может вносить изменения в товары через приложение, есть возможность добавить товар, редактировать и удалить.

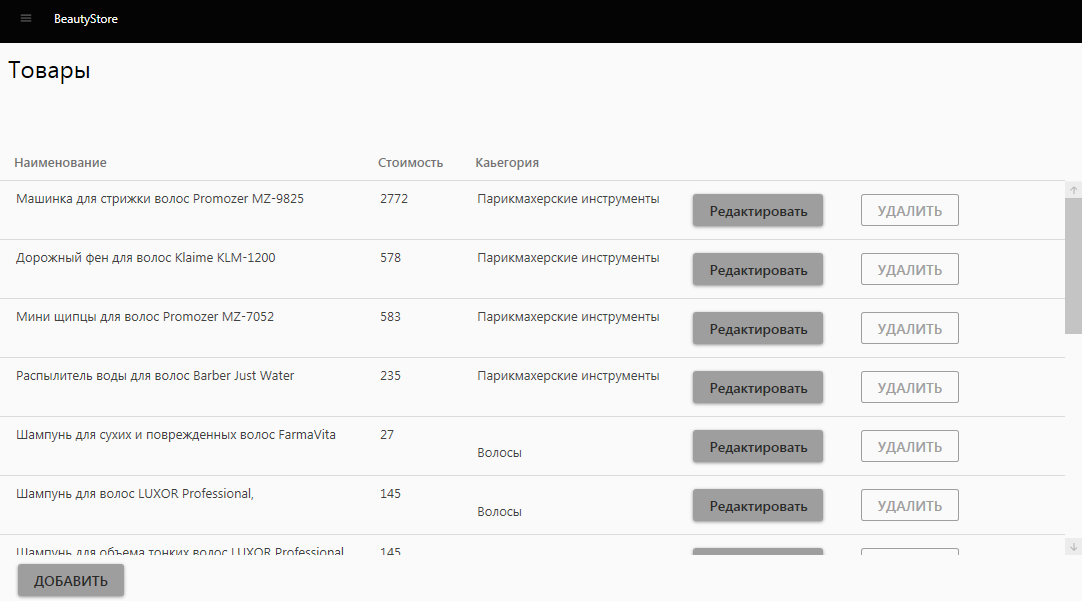


Рисунок 31 – Редактирование и удаления товаров

Создание формы добавления товаров показано на рисунке показано на рисунке 32.

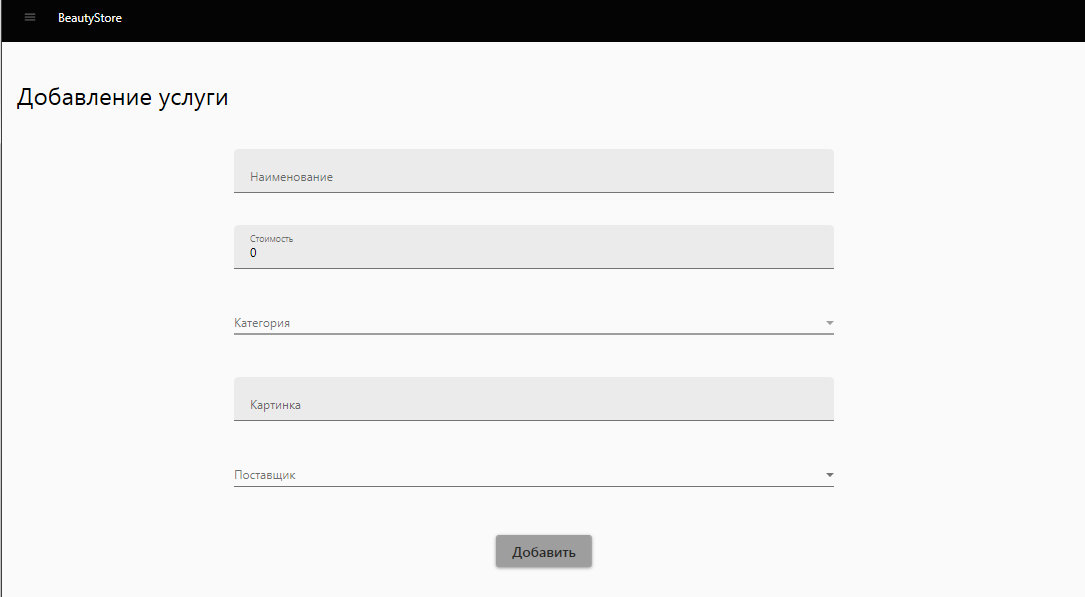


Рисунок 32 – Форма добавления товаров

Создание формы редактирование товаров, показано на рисунке показано на рисунке 33.

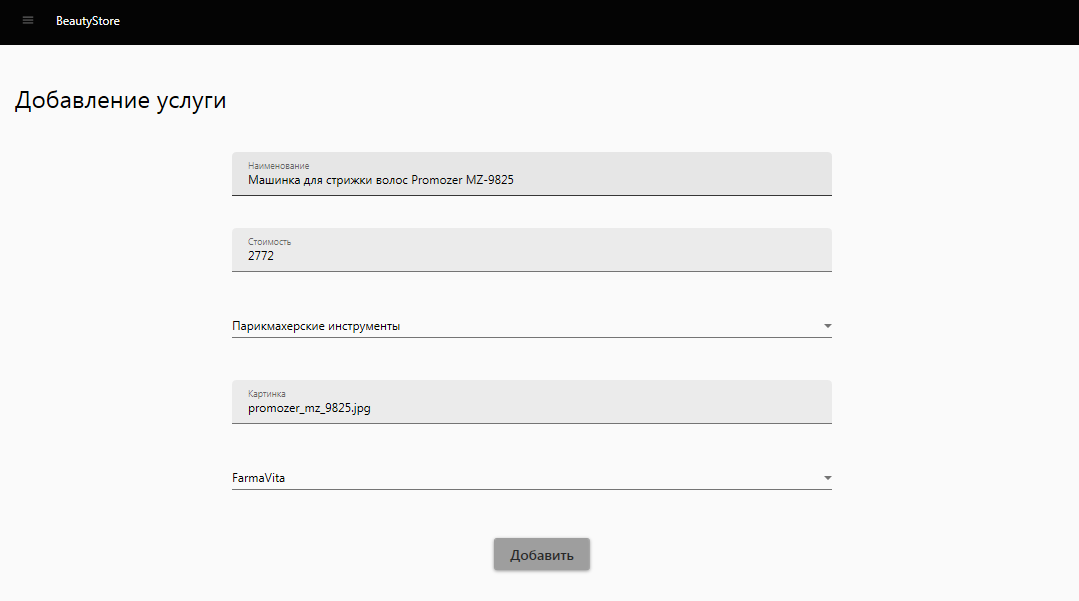


Рисунок 33 – Форма редактирования товаров

Создание формы удаления товара показано на рисунке показано на рисунке 34.

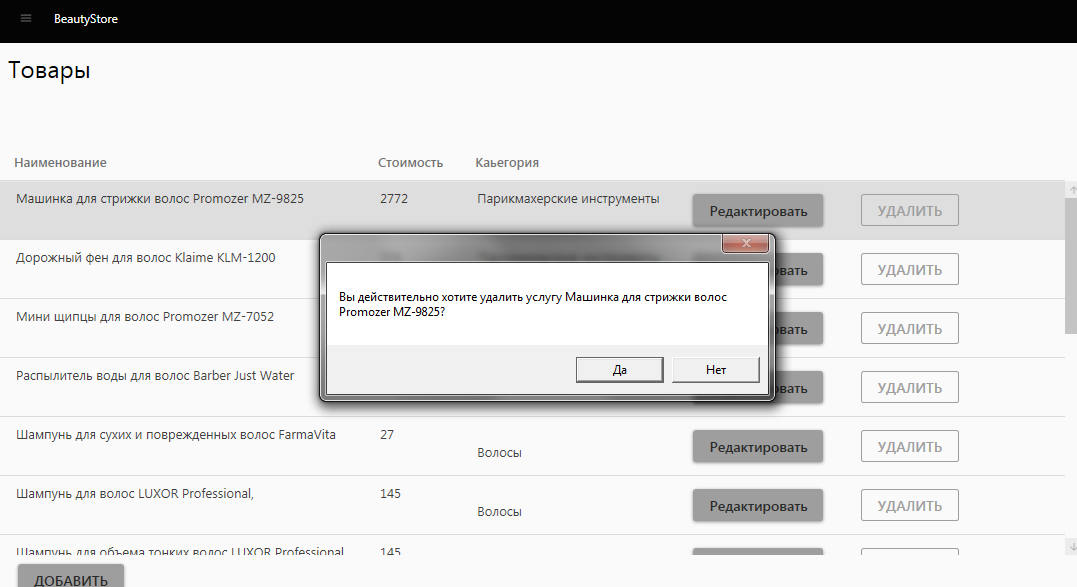


Рисунок 34 Форма удаления товаров

3. СТРАТЕГИЯ ПРОДВИЖЕНИЯ

## 3.1 Маркетинг

## SEO

# ТЕСТИРОВАНИЕ

# ЗАКЛЮЧЕНИЕ

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. ГОСТ 19.101–77 Единая система программной документации (ЕСПД). Виды программ и программных документов

2. ГОСТ 19.201–78 Единая система программной документации (ЕСПД). Техническое задание. Требования к содержанию и оформлению

3. ГОСТ 19.402–78 Единая система программной документации (ЕСПД). Описание программы

4. ГОСТ 19.404–79 Единая система программной документации (ЕСПД). Пояснительная записка. Требования к содержанию и оформлению

5. Документация PHP – <https://www.php.net/manual/ru/index.php>

6. Документация MySQL – <https://dev.mysql.com/doc/>

7. Документация HTML и CSS – http://htmlbook.ru/html

8. Актуальность развития малого бизнеса [Электронный ресурс] – 2019 г. – http://conf.omua.ru/content/aktualnost–razvitiya–malogo–biznesa–v–rossii

9. Губарев А.А. Разработка баз данных / А.А. Губарев. – СПб.: Питер, 2019. – 564 с.

10. Сомин, А.В. Разработка баз данных/ А.В. Сорокин. – Москва, –2017. – 259 с.

11. Основы UML – диаграммы использования [Электронный ресурс] – 2019 г. – [https://pro–prof.com/archives/2594](https://pro-prof.com/archives/2594)

12. Основные этапы создания информационной системы [Электронный ресурс] – 2019 г. – http://itconcord.ru/articles/stages/

13. Грейди, Буч. UML. Руководство пользователя / Грейди Буч, Джеймс Рамбо, Айвар Д. – М: Питер, ДМК пресс, 2018. – 436 с.

14. Настройка приложений баз данных / Б. Новиков, Г. Домбровская – Издательство БХВ, 2006 г. – 252 c.

15. Баканов, М.И. Теория проектирования баз данных: Учебник / М.И. Баканов, А.Д. Шеремет. – М.: Финансы и статистика, 2019. – 265 с.

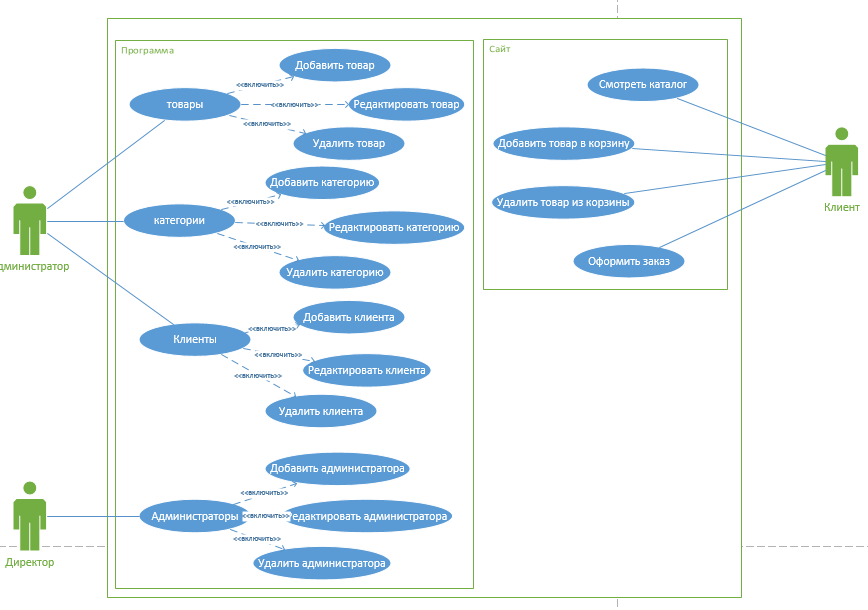
16.Кузнецов, М. Самоучитель MySQL / М. Кузнецов, И. Симдянов. – СПб:БХВ-Петербург, 2018 – 546 с.

17. Информационные системы: Учебник для вузов. 3-е издание. / Ю.С.Избачков — Издательский дом «Питер», 2010 г. — 450 с.

18. Мюллер, Р.Дж. Базы данных и UML. Проектирование / Р.Дж. Мюллер. - М.: ЛОРИ, 2002. - 420 c.

# Приложение А

Диаграмма использования



# Приложение Б

Диаграмма деятельности

# Приложение В

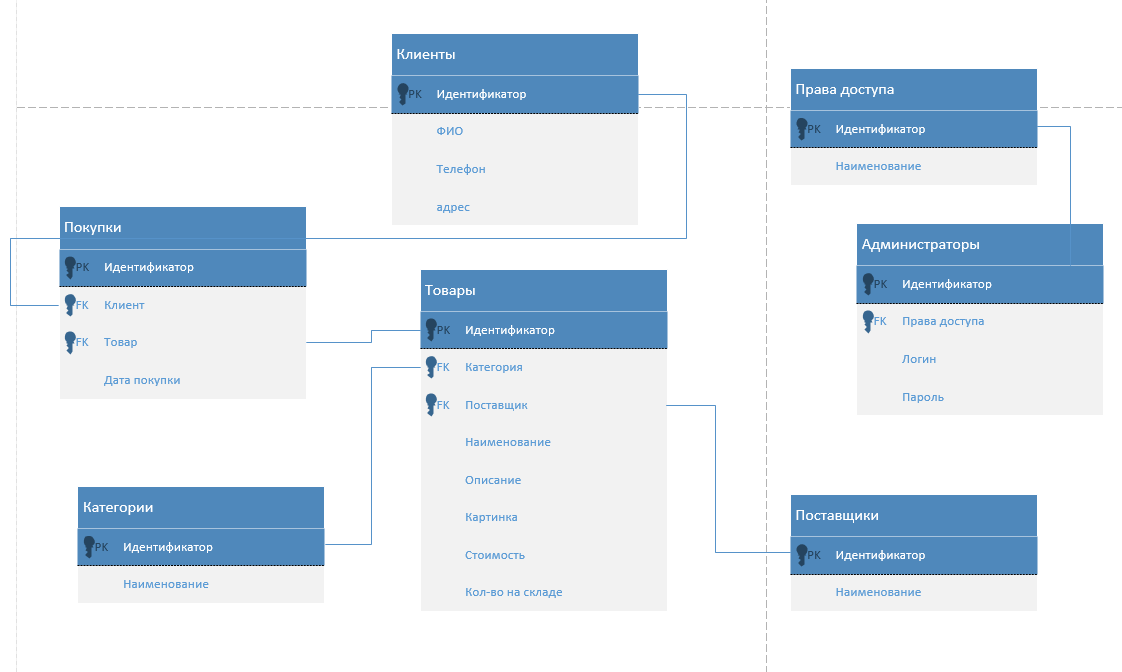
Физическое проектирование

# Приложение Г

Семантическая модель

# Приложение Д

Логическая модель



# Приложение Е

Проектирование интерфейса

# Приложение Ё

Создание страницы каталога

# Приложение Ж

Создание страницы редактирования категорий