**МИНОБРНАУКИ РОССИИ**

**«Юго-Западный государственный университет»**

**Кафедра «Вычислительная техника»**

**КУРСОВАЯ РАБОТА**

по дисциплине «Управление данными»

на тему «Проектирование базы данных автосалона»

Направление подготовки 09.03.02 Информационные системы и технологии

Автор работы Романов Г.А. \_\_\_\_\_\_\_\_\_\_\_\_

фамилия, инициалы подпись, дата

Группа ИС-91б

Руководитель работы Бобынцев Д.О. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

фамилия, инициалы подпись, дата

Работа защищена \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Члены комиссии \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Бобынцев Д.О.

подпись, дата фамилия, инициалы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Киселёв А.В.

подпись, дата фамилия, инициалы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ткаченко А.В.

подпись, дата фамилия, инициалы

Курск 2022 г.

**МИНОБРНАУКИ РОССИИ**

**«Юго-Западный государственный университет»**

Кафедра «Вычислительная техника»

ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Студента Романова Георгий Александровича шифр 19-06-0146 группа ИС-91б

1.Тема «Проектирование базы данных автосалона»  
2. Срок представления проекта к защите «12 января» 2022г.

3. Исходные данные для проектирования.

4. Содержание пояснительной записки курсовой работы:

4.1. Анализ предметной области

4.2. Описание разработки

4.3. Заключение

4.4. Список использованной литературы

4.5. Листинг программы

5. Перечень графического материала: отсутствует

Руководитель работы (проекта) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.О. Бобынцев

подпись, дата инициалы, фамилия

Задание принял к исполнению Г.А. Романов

подпись, дата инициалы, фамилия

**РЕФЕРАТ**

Курсовая работа: «Проектирование базы данных автосалона» состоит из двух глав, введения и заключения общим объемом 70 страниц. В работе использовано 5 источников литературы.

Объектом исследования является процесс реализации базы данных автосалона и клиентского приложения.

Предметом исследования в работе являются методы и инструментальные среды разработки, отладки и тестирования программного продукта.

Цель курсовой работы – это оптимизация информационных процессов автосалона посредством разработки базы данных и клиентского приложения.

Техническая задача курсовой работы является разработка базы данных автосалона и клиентского приложения. Задачами курсовой работы являются:

* Анализ подходов к реализации базы данных автосалона и клиентского приложения;
* Выбор инструментальной среды и разработка модели данных приложения;
* Анализ полученных результатов от использования клиентского приложения.

В работе дано описание создания базы данных и клиентского приложения, позволяющего управлять продажами автомобилей в автосалоне. Разработана модель данных. Приложение позволит владельцу и персоналу легко управлять продажами автомобилей.

**Содержание**

[Введение 5](#_Toc88311333)

[1Анализ предметной области 6](#_Toc88311334)

[1.1Описание предметной области 6](#_Toc88311335)

[1.2 Выделение сущностей 8](#_Toc88311336)

[1.3 Описание связей сущностей 11](#_Toc88311337)

[1.4 Язык структурированных запросов SQL 13](#_Toc88311338)

[2Описание разработки 16](#_Toc88311339)

[2.1 Переход к реляционной модели и создание базы данных 16](#_Toc88311340)

[2.2 Создание клиентского приложения 21](#_Toc88311341)

[Заключение 33](#_Toc88311342)

[Список использованных источников 34](#_Toc88311343)

[Листинг программы 35](#_Toc88311344)

# **Введение**

Данная курсовая работа имеет целью разработку базы данных и клиентского приложения, позволяющей пользователю оформить продажу автомобиля.

Многие действия при оформлении продажи автомобиля являются цикличными и требующими внимательности. Поэтому часто случается, что при повторении одних и тех же действий, сотрудник начинает совершать ошибки при оформлении, которые занимают время как сотрудника, так и покупателя. Для решения этих проблем будет использоваться клиентское приложение, подключённое к базе данных автосалона, которое позволит автосалону ускорить свою работу в разы, за счёт ячеек с выбором варианта при оформлении.

Цель работы – разработка базы данных и клиентского приложения для автосалона.

Для достижения поставленной цели следует решить следующие задачи:

1. изучить и описать среду разработки базы данных и клиентского приложения;
2. разработать приложение для управления продажами автосалона.
3. проанализировать работу разработанного приложения.

В качестве СУБД использовался Microsoft SQL Server 2019.

# **Анализ предметной области**

# **Описание предметной области**

Вид деятельности автосалона – продажа автомобилей. Некоторая информация о продажах в автосалоне:

1. Уполномоченный сотрудник (менеджер или оператор) автосалона вносит в специальную программу данные, которые не будут меняться или же меняться редко. Например, сведения о производителях автомобилей. После этого, данные будут отправлены в БД.
2. Уполномоченный сотрудник при поступлении автомобиля в автосалон записывает сведения о прибывшем автомобиле и заносит их БД.
3. Каждый автомобиль, находящийся в наличии, должен быть охарактеризован некоторыми данными: производителем, моделью, стоимостью, цветом.
4. При оформлении продажи автомобиля, уполномоченное лицо обязано занести сведения о покупателе в БД. Необходимыми сведениями являются паспортные данные покупателя.

Деятельность автосалона заключается в закупке автомобилей у производителей и продажа физическим и юридическим лицам. За счет разницы цены покупки и продажи автомобилей автосалон ведет свою деятельность.

В автосалон автомобили попадают либо от непосредственного производителя, либо от поставщика, занимающегося перепродажей авто. Доставкой автомобиля со склада производителя или поставщика до автосалона занимаются специально обученные люди – перегонщики. Перегонщики не являются персоналом автосалона, они могут работать либо у производителя, либо у поставщика, либо быть аутсорсингом. Хранение в базе данных обуславливается тем, что в случае нахождения проблем в автомобиле, знать кто его доставил. Их задача привести авто в целости и сохранности. После этого авто добавляется в БД автосалона и может быть продано покупателю. Покупатель может либо сделать заказ на автомобиль, что бы автосалон заказал автомобиль у производителя или перекупщика, либо зарезервировать.

Перейдём к следующему этапу проектирования БД.

## **1.2 Выделение сущностей**

Исходя из вышеупомянутого анализа предметной области, выделим сущности для базы данных:

* Перегонщики;
* Заказы;
* Поставщики;
* Автомобили;
* Покупатели;
* Сотрудники;
* Модель;
* Производители.

Сущность «Перегонщики» содержит такие атрибуты как:

* Собственный идентификатор;
* ФИО;
* Паспортные данные.

Объект «Заказы» имеет следующие свойства:

* Собственный идентификатор;
* Дата заказа;
* Идентификатор клиента;
* ФИО клиента;
* Идентификатор автомобиля;
* Наименование автомобиля;
* Сумма оплаты;
* Статус оплаты;
* Идентификатор сотрудника;
* ФИО сотрудника.

Сущность «Поставщики» содержит атрибуты:

* Собственный идентификатор;
* ИНН;
* Наименование;
* Реквизиты банка;
* Адрес.

«Автомобили» содержит такие атрибуты как:

* Идентификатор автомобиля;
* Дата поставки;
* Наименование модели;
* Цена;
* Объём двигателя;
* Номер двигателя;
* Идентификатор перегонщика;
* ФИО перегонщика;
* Идентификатор поставщика;
* Наименование поставщика;
* Статус наличия.

Объект «Покупатели» содержит такие атрибуты:

* Собственный идентификатор;
* ФИО;
* Адрес;
* Паспортные данные.

Сущность «Сотрудники» включает такие свойства:

* Собственный идентификатор;
* ФИО;
* Занимаемая должность;
* Номер телефона;
* Пол;
* Адрес.

Объект «Модель» имеет такие атрибуты как:

* Собственный идентификатор;
* Марка;
* Тип кузова;
* Коробка передач;
* Наименование модели;
* Год выпуска;
* Цвет;
* Идентификатор производителя;
* Наименование производителя.

Объект «Производители» получил такие атрибуты как:

* Собственный идентификатор;
* Наименование;
* Страна.

У каждой сущности есть идентификатор ID. Это важно, так как такой подход обеспечивает надёжность, путём исключения варианта, когда сущности имеют одинаковые атрибуты. Идентификатор является для каждого отдельного объекта уникальным.

Зная сущности БД их надо соединить с помощью специальных связей, которые описываются дальше.

## **1.3 Описание связей сущностей**

В любой реляционной БД имеется хотя бы один из трёх типов связи:

* Один-ко-многим;
* Один-к-одному;
* Многие-ко-многим.

Один-ко-многим- характеризуется тем, что запись в одной таблице может быть связана с множеством записей другой таблице (Рис.1). В реляционной модели базы данных эта связь использует две таблицы.

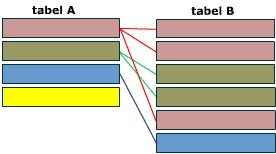


Рисунок 1 - Связь один-ко-многим

Многие-ко-многим имеет три таблицы. Две таблицы – источника и одна соединительная таблица. Первичный ключ соединительной таблицы – составной. Она состоит из двух полей, двух внешних ключей, которые ссылаются на первичные ключи первой и второй таблиц (Рис.2).

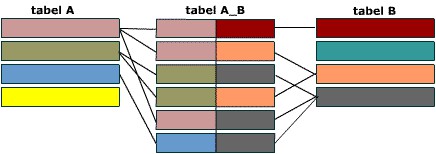


Рисунок 2 - Связь многие-ко-многим

Связь один-к-одному легко моделируется в одной таблице. Записи таблицы содержат данные, которые находятся в связи один-к-одному с первичным ключом или записью.

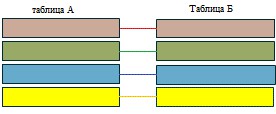


Рисунок 3 – Связь один-к-одному

Имея схему строения будущей БД и зная какую информацию, необходимо будет вносить и получать из БД, перейдём к реализации проекта.

## **1.4 Описание ER-модели БД**

В ER-модели сущности обычно имеют логические связи, обусловленные предметной областью. Связи имеют функциональную и количественную характеристику. Функциональная характеристика описывает саму суть связи между сущностями. Виды каждой связи в ER-модели определяются аналитически при рассмотрении заданной предметной области. Окончательная модель должна содержать все выделенные сущности с необходимым набором связей и характеристиками связей.

Определим связи между описанными сущностями.

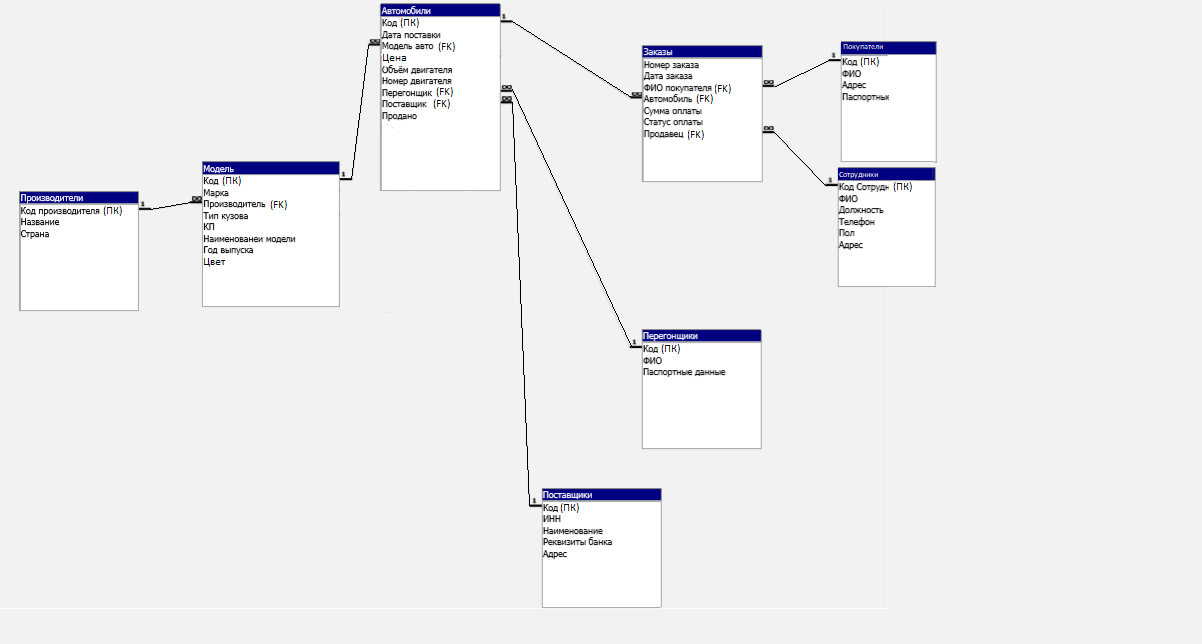


Рисунок 4 – ER-модель БД

У одного производителя может быть много различных моделей. Но у конкретной модели есть только один производитель. Поэтому в данном случае имеет место связь 1:М – один производитель, много продукции.

Одна модель может быть у различных автомобилей, но у каждого автомобиля есть только одна модель. Поэтому в данном случае имеет место связь 1:М – одна модель, множество авто.

Один поставщик может продать множество различных автомобилей, но у одного автомобиля будет только один поставщик. Поэтому в данном случае имеет место связь 1:М – один поставщик, много проданных авто.

Один перегонщик может привести много разных автомобилей, но один автомобиль может привести только один перегонщик. Поэтому в данном случае имеет место связь 1:М – один перегонщик, много привезённых авто.

Один автомобиль может быть в разных заказах, ведь покупатель может заказать автомобиль или зарезервировать. Но в одном заказе может быть только один автомобиль. Поэтому в данном случае имеет место связь 1:М – один авто, много заказов.

Один покупатель может сделать множество заказов, но у одного заказа может быть только один покупатель. Поэтому в данном случае имеет место связь 1:М – один покупатель, много заказов.

Один сотрудник может оформить множество заказов, но у заказа будет только один сотрудник, оформивший сделку.

## **1.5 Язык структурированных запросов SQL**

SQL — Структурированный Язык Запросов — это стандартный язык запросов по работе с реляционными базами данных.

Существует две разновидности языка SQL: PL-SQL и T-SQL. Первая используется в таких СУБД как Oracle и MySQL, вторая же применяется в SQL Server.

T-SQL может принадлежать к одному из следующих типов:

DDL (Data Definition Language / Язык определения данных). К этому типу относятся различные команды, которые создают базу данных, таблицы, индексы, хранимые процедуры и т.д. В общем определяют данные.

В частности, к этому типу мы можем отнести следующие запросы:

• CREATE: создает объекты базы данных (саму базу данных, таблицы, индексы и т.д.)

• ALTER: изменяет объекты базы данных

• DROP: удаляет объекты базы данных

• TRUNCATE: удаляет все данные из таблиц

DML (Data Manipulation Language / Язык манипуляции данными). К этому типу относят команды на выбор данных, их обновление, добавление, удаление - в общем все те команды, с помощью которыми мы можем управлять данными.

К этому типу относятся следующие запросы:

• SELECT: извлекает данные из БД

• UPDATE: обновляет данные

• INSERT: добавляет новые данные

• DELETE: удаляет данные

# **Описание разработки**

## **Переход к реляционной модели и создание базы данных**

Перейдём к созданию реляционной модели. Для начала определим все сущности, а именно «Перегонщики», «Заказы», «Поставщики», «Автомобили», «Покупатели», «Сотрудники», «Модель», «Производители» в виде таблиц. Их названия, следующие:

* Перегонщики;
* Заказы;
* Поставщики;
* Автомобили;
* Покупатели;
* Personal (сотрудники);
* Модель;
* Производители;

Таблица «Покупатели» содержит такие столбцы как (Рис. 4):

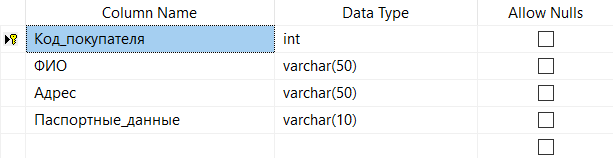


Рисунок 4 - Вид таблицы «Покупатели» в конструкторе

Из данного рисунка можно заметить, что ни одно поле не допускает значение null.

Поле «Паспортные\_данные» имеет свойство Unique.

Таблица «Автомобили» содержит следующие атрибуты (Рис. 5):

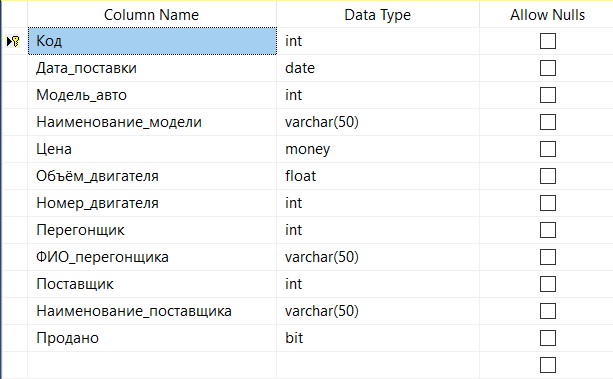


Рисунок 5 - Вид таблицы «Автомобили» в конструкторе

Поле «Номер\_двигателя» имеет свойство Unique.

Таблица «Покупатели» содержит такие столбцы как (Рис. 6):

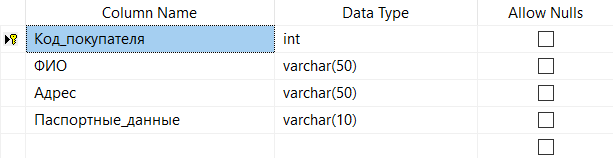


Рисунок 6 - Вид таблицы «Покупатели» в конструкторе

Поле «Паспортные\_данные» имеет свойство Unique.

Таблица «Модель» содержит такие атрибуты (Рис. 7):

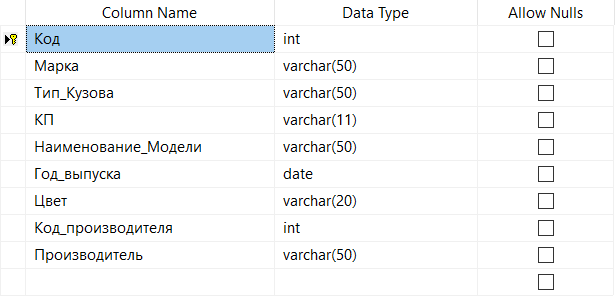


Рисунок 7 - Вид таблицы «Модель» в конструкторе

Поле «КП» может иметь значения «Механика», «Автомат» или «Полуавтомат».

Таблица «Перегонщики» содержит такие атрибуты (Рис. 8):

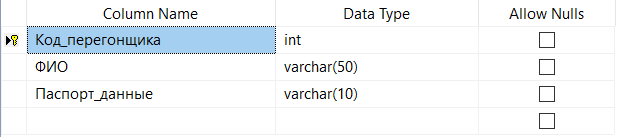


Рисунок 8 - Вид таблицы «Перегонщики» в конструкторе

Поле «Паспорт\_данные» имеет свойство Unique.

Таблица «Заказы» содержит такие атрибуты (Рис. 9):

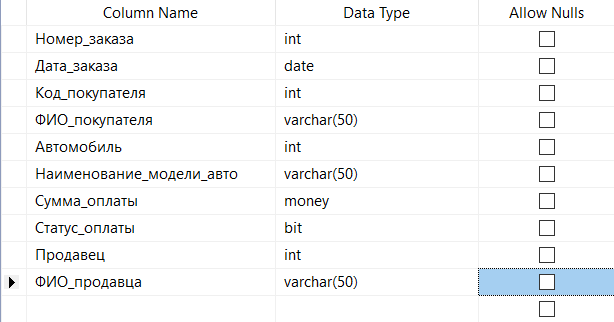


Рисунок 9 - Вид таблицы «Заказы» в конструкторе

Поле «Номер\_заказа» имеет свойство Unique.

Таблица «Поставщики» содержит такие атрибуты (Рис. 10):

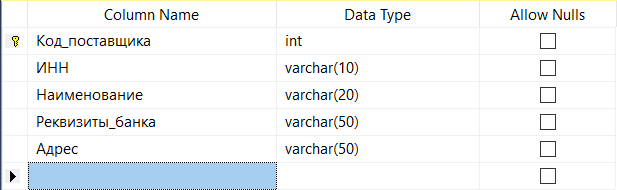


Рисунок 10 - Вид таблицы «Поставщики» в конструкторе

Поле «ИНН» имеет свойство Unique.

Таблица «Personal» (сотрудники) содержит такие атрибуты (Рис. 11):

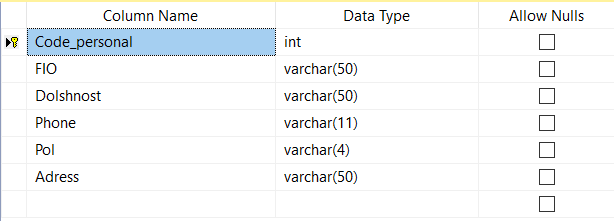


Рисунок 11 - Вид таблицы «Поставщики» в конструкторе

Поле «Phone» имеет свойство Unique. Поле «Pol» может иметь значения «Муж» или «Жен».

Таблица «Производители» содержит такие атрибуты (Рис. 12):

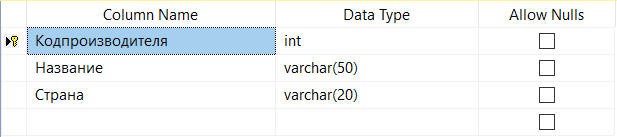


Рисунок 12 - Вид таблицы «Производители» в конструкторе

Поле «Название» имеет свойство Unique.

Таблица 1. Отношения сущность-связь

|  |  |
| --- | --- |
| **Название** | **Описание** |
| FK\_Производители\_ToМодель | ON DELETE NO ACTION ON UPDATE CASCADE |
| FK\_Модель\_ToАвтомобили | ON DELETE NO ACTION ON UPDATE CASCADE |
| FK\_Автомобили\_ToЗаказы | ON DELETE NO ACTION ON UPDATE CASCADE |
| FK\_Поставщики\_ToАвтомобили | ON DELETE NO ACTION ON UPDATE CASCADE |
| FK\_Перегонщики\_ToАвтомобили | ON DELETE NO ACTION ON UPDATE CASCADE |

Продолжение таблицы 1:

|  |  |
| --- | --- |
| FK\_Покупатели\_ToЗаказы | ON DELETE NO ACTION ON UPDATE CASCADE |
| FK\_Сотрудники\_ToЗаказы | ON DELETE NO ACTION ON UPDATE CASCADE |

После определения сущностей будущей базы данных, создадим диаграмму базы данных, чтобы точно знать, какая связь имеется среди сущностей. Диаграмма базы данных выглядит следующим образом.

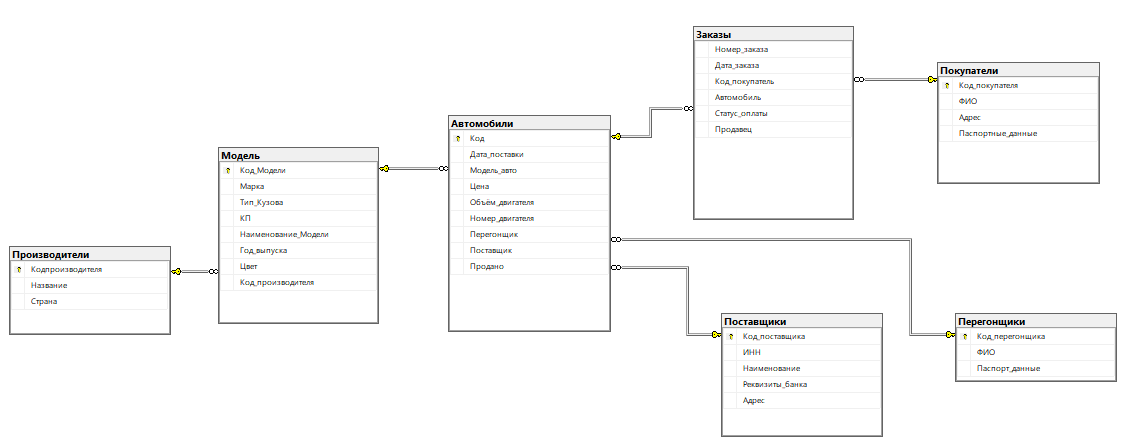


Рисунок 13 - Диаграмма базы данных

После создания БД перейдём к созданию клиентского приложения.

## **2.2 Создание клиентского приложения**

Клиентское приложение реализовано с помощью среды разработки Visual Studio 2019, а также при помощи технологий WinForms, MS SQL Server и ADO.NET.

WinForms – это достаточно популярный на данный момент шаблон для создания форменных приложений для персонального компьютера.

MS SQL Server – это, как уже упоминалось, система управления реляционными базами данных, использующее язык SQL и его процедурное расширение Transact-SQL.

Microsoft.Data.SqlClient – Пространство имен [System.Data.SqlClient](https://docs.microsoft.com/ru-ru/dotnet/api/system.data.sqlclient?view=netframework-4.8) является поставщиком данных платформы .NET для SQL Server. Он нужен для работы с БД.

При открытии приложения открывается главная форма (Рис. 14).

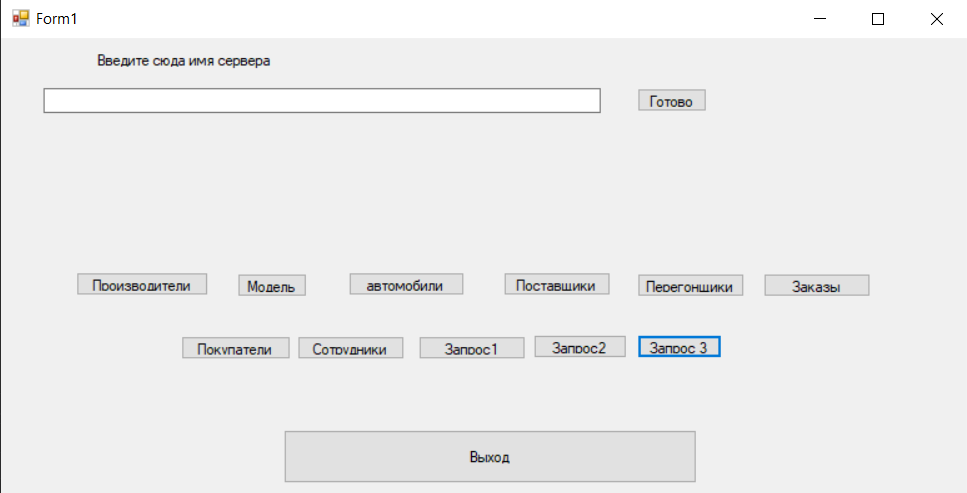


Рисунок 14 – Главная форма приложения

Здесь можно выбрать нужную таблицу нажав на соответствующую кнопку. Поле для ввода имени сервера используется для ввода имени сервера на локальной машине, это необходимо для правильного подключения клиентского приложения к БД.

После нажатия на кнопку «Производители» откроется окно с данными по производителям.

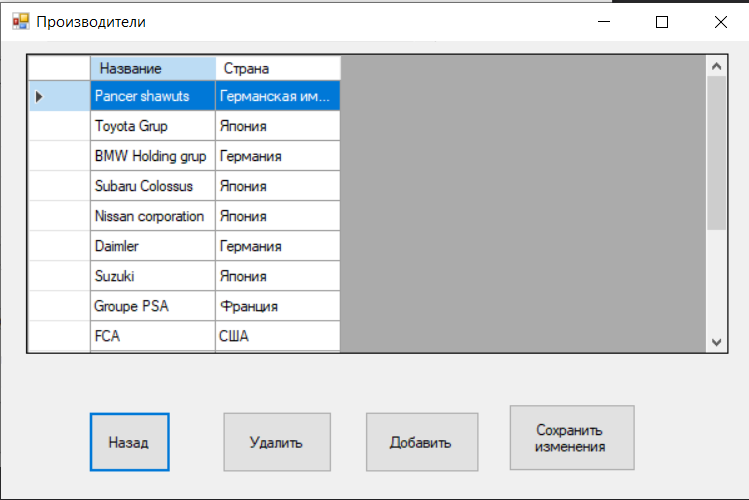


Рисунок 15 – Форма с таблицей «Производители»

Кнопка «Назад» возвращает в главную форму. Кнопка «Удалить» удаляет выделенную строку. Строки нужно выделять компьютерной мышью. Изменить текст можно прямо в таблице, выбрав нужную строку. Кнопка «Добавить» добавит новую строку, которую необходимо заполнить. Для сохранения результата работы необходимо нажать кнопку «Сохранить изменения», тогда данные в БД обновятся.

После нажатия кнопки «Модель», в главной форме, откроется окно с данными по моделям авто.

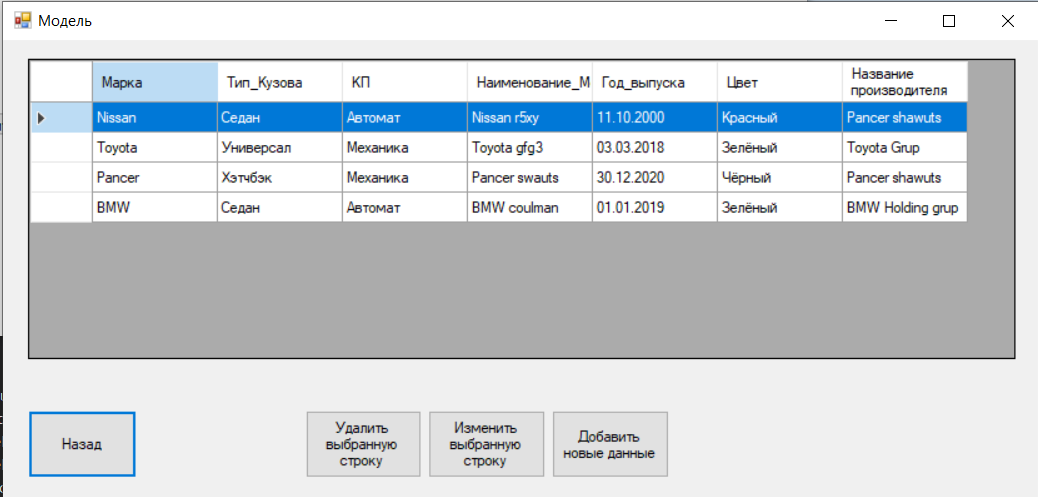


Рисунок 16 – Форма с таблицей «Модель»

Здесь кнопки «Назад», «Удалить выбранную строку» и «Сохранить изменения» аналогичны предыдущей форме.

Кнопка «Изменить выбранную строку» открывают специальную форму для изменения строки.

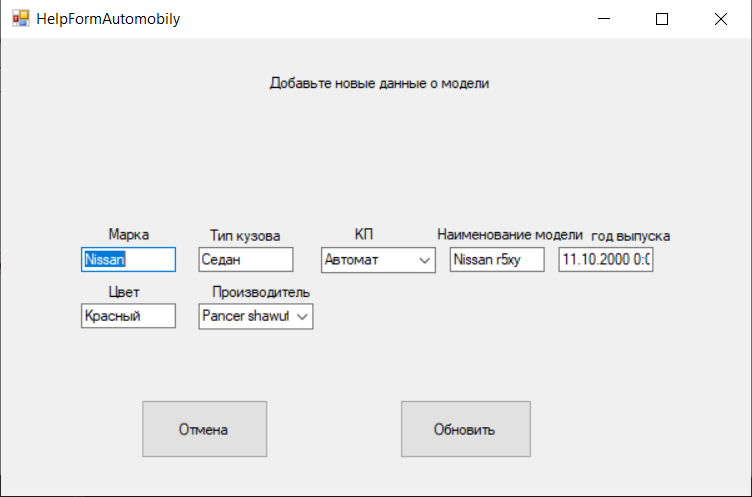


Рисунок 17 – Форма для изменения данных о модели

После ввода новых данных нужно нажать кнопку «Обновить», тогда данные обновятся в БД.

Кнопка «Добавить новые данные» открывают специальную форму для добавления новой строки.

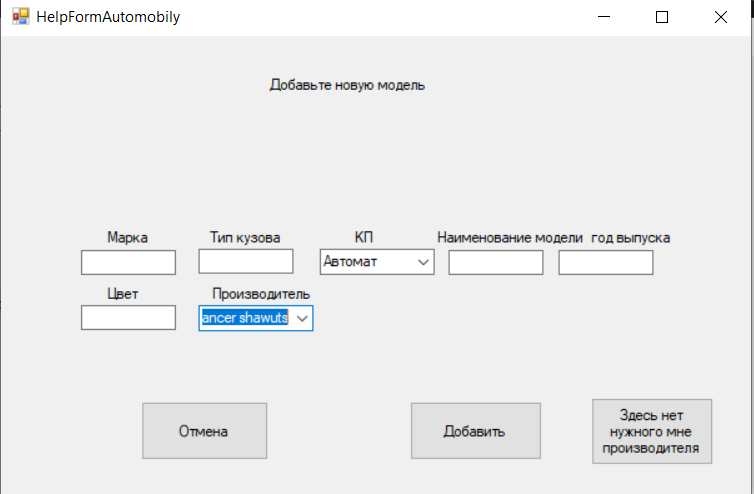


Рисунок 18 – Форма для ввода данных о модели

После ввода новых данных нужно нажать кнопку «Добавить», тогда данные обновятся в БД. Если нужного производителя нет в наличии, то можно

добавить нового. Для этого нужно нажать кнопку «Здесь нет нужного мне производителя». После этого откроется форма для добавления нового производителя.

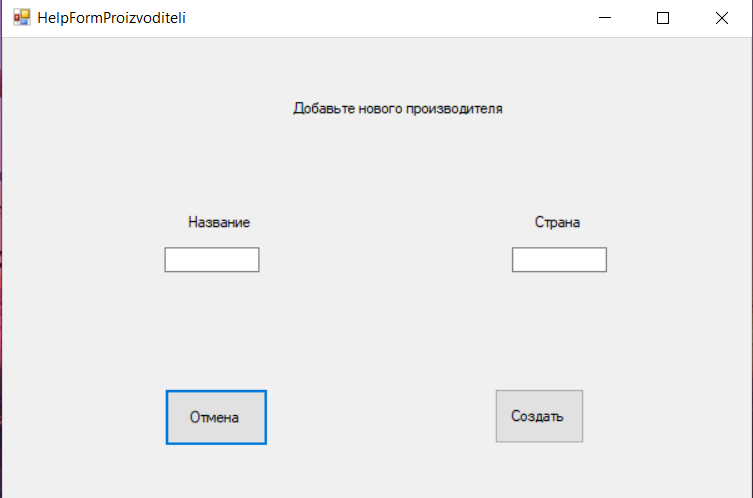


Рисунок 19 – Форма для ввода данных о производителе

Кнопка «Отмена» возвращает обратно, кнопка «Создать» добавляет нового производителя в БД.

После нажатия кнопки «Автомобили», в главной форме, откроется окно с данными по автомобилям.

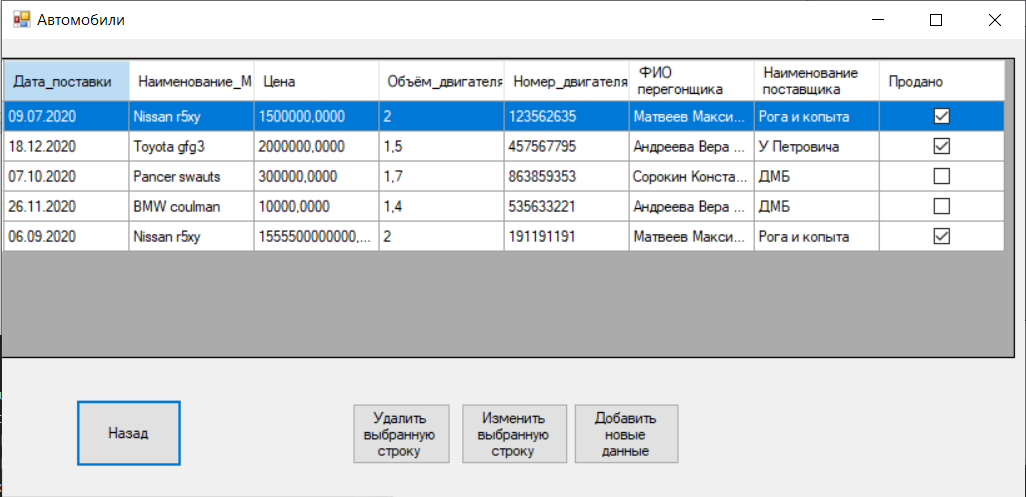


Рисунок 20 – Форма с таблицей «Автомобили»

Кнопки «Назад», «Удалить выбранную строку» и «Сохранить изменения» аналогичны описанным в предыдущей форме. Кнопки «Изменить выбранную строку» и «Добавить новые данные» открывают специальные окна для добавления или редактирования данных.

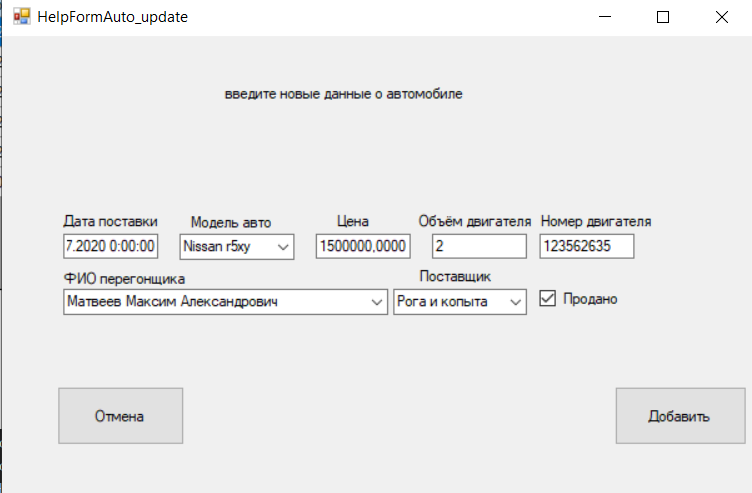


Рисунок 21 – Форма для изменения данных о автомобиле

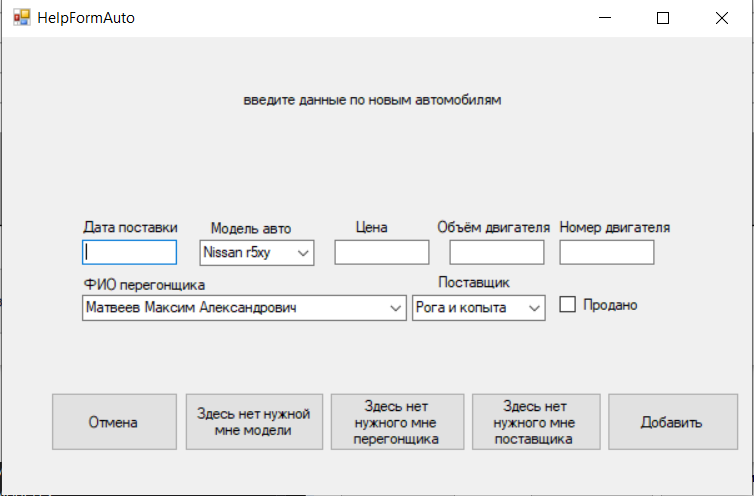


Рисунок 22 – Форма для ввода нового автомобиля

Кнопки «Отмена» и «Добавить» аналогичны предыдущим.

Кнопки «Здесь нет нужной мне модели», «Здесь нет нужного мне перегонщика», «Здесь нет нужного мне поставщика» нужны для добавления новых данных на случай, если будет отсутствовать нужные. Они переносят на специальные формы, где можно добавить новые данные по выбранной категории данных.

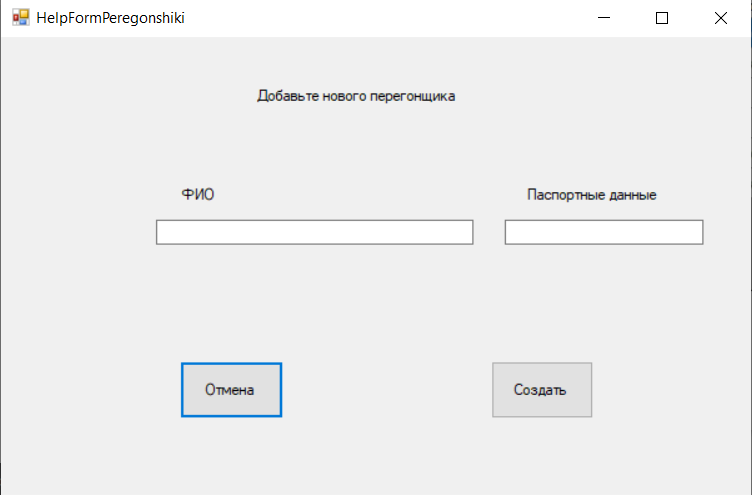


Рисунок 23 – Форма для ввода нового перегонщика

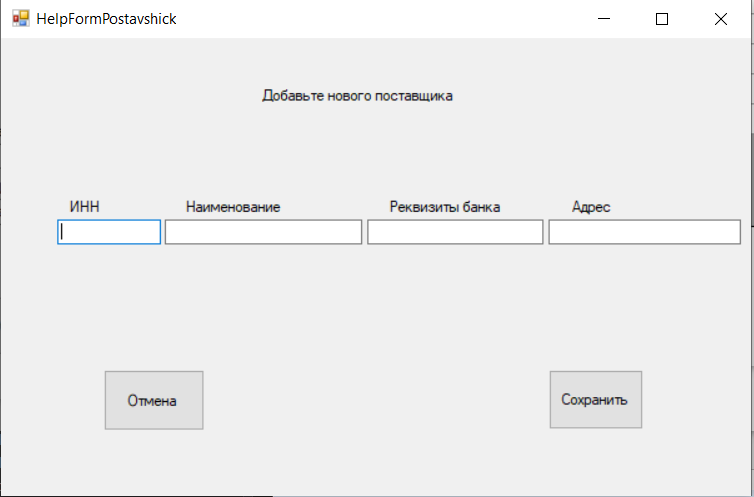


Рисунок 24 – Форма для ввода нового поставщика

Кнопки «Отмена» и «Сохранить» аналогичны вышеописанным.

Форма добавления новой модели уже была описана выше.

После нажатия кнопки «Поставщики», в главной форме, откроется окно с данными по поставщикам.

.

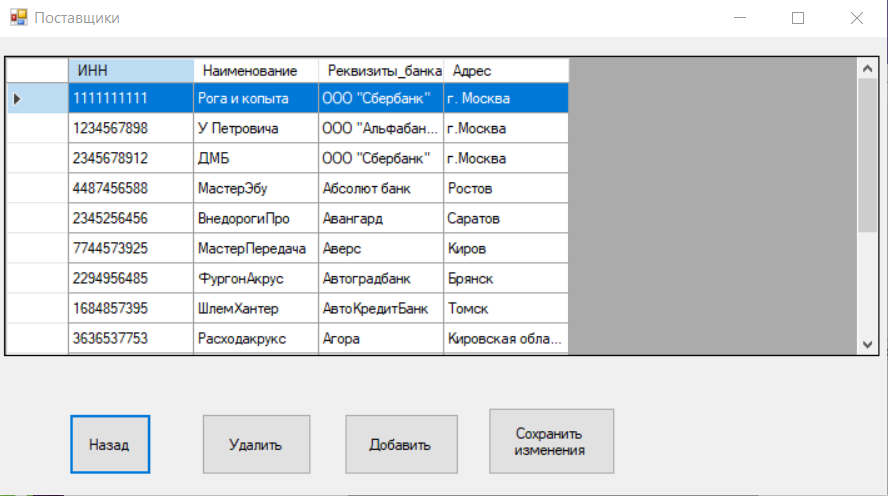


Рисунок 25 – Форма с таблицей «Поставщики»

После нажатия кнопки «Перегонщики», в главной форме, откроется окно с данными по перегонщикам.

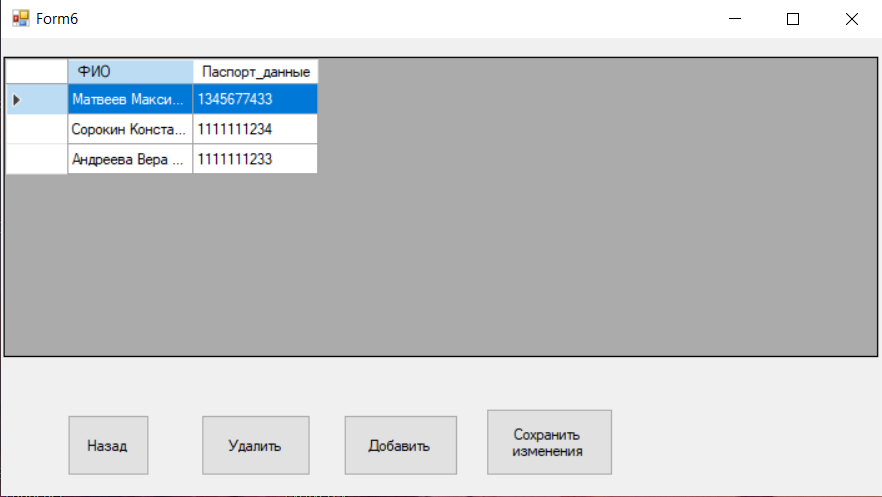


Рисунок 26– Форма с таблицей «Перегонщики»

Кнопки «Назад», «Удалить» и «Сохранить изменения» ничем не отличаются от предыдущих. Кнопка «Добавить» больше не открывает новое

окно, а создаёт новую строку в таблице.

После нажатия кнопки «Заказы», в главной форме, откроется окно с данными по заказам.

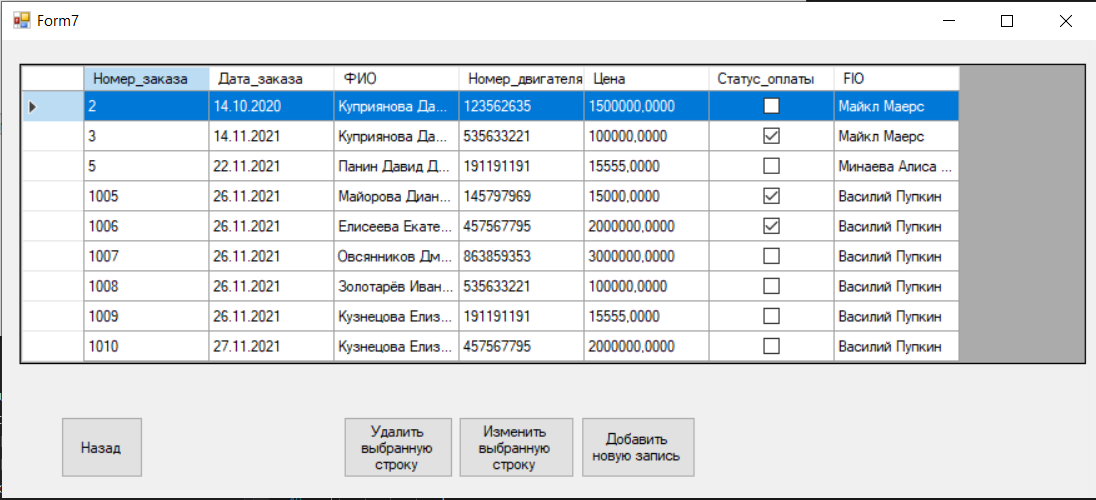


Рисунок 27– Форма с таблицей «Заказы»

Кнопки «Изменить выбранную строку» и «Добавить новую запись» открывают специальные окна для добавления или редактирования данных.

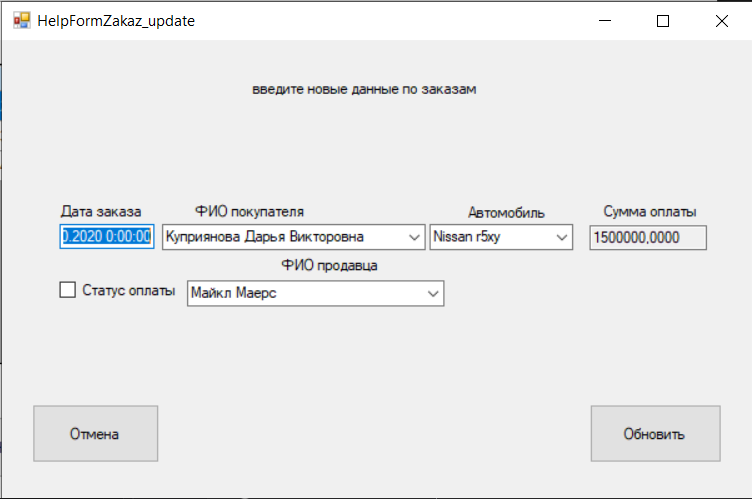


Рисунок 28– Форма редактирования данных о заказе

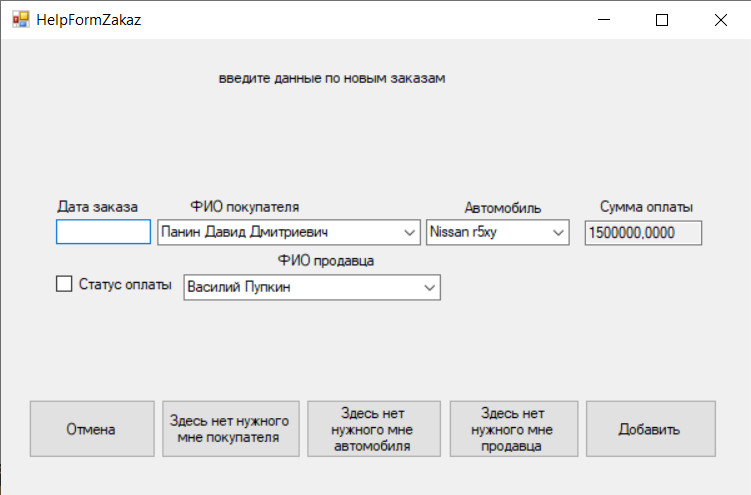


Рисунок 29– Форма добавления нового заказа

Кнопки «Здесь нет нужного мне покупателя», «Здесь нет нужного мне автомобиля» и «Здесь нет нужного мне продавца» открывают специальные формы добавления данных.

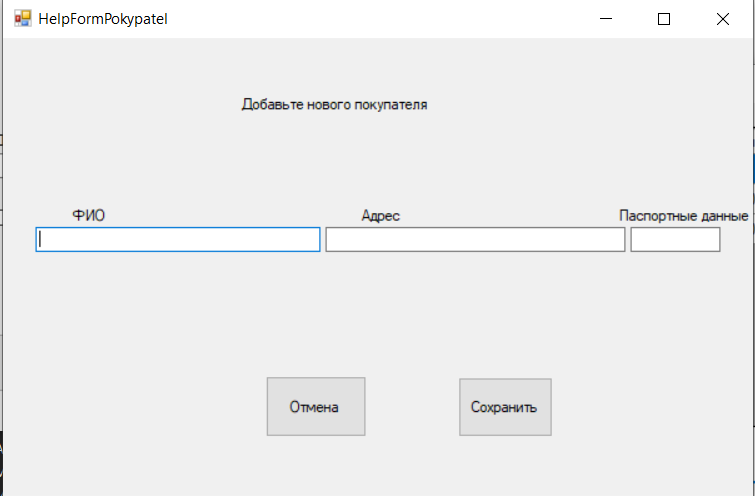


Рисунок 30– Форма добавления нового покупателя

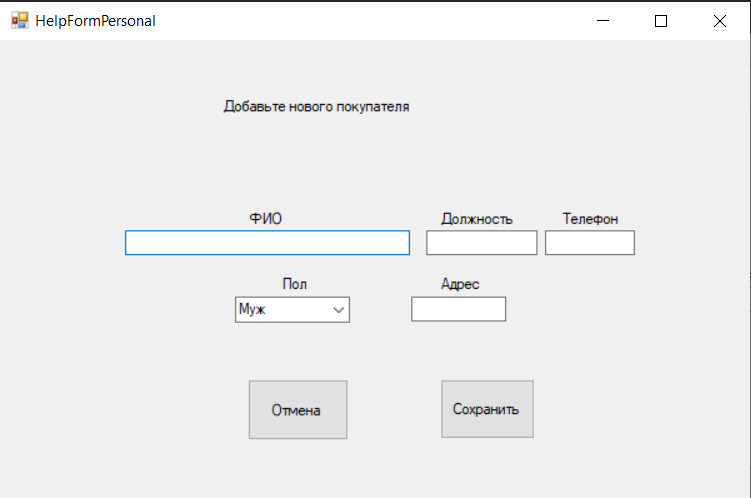


Рисунок 31– Форма добавления нового сотрудника

После нажатия кнопки «Покупатели», в главной форме, откроется окно с данными по покупателям.

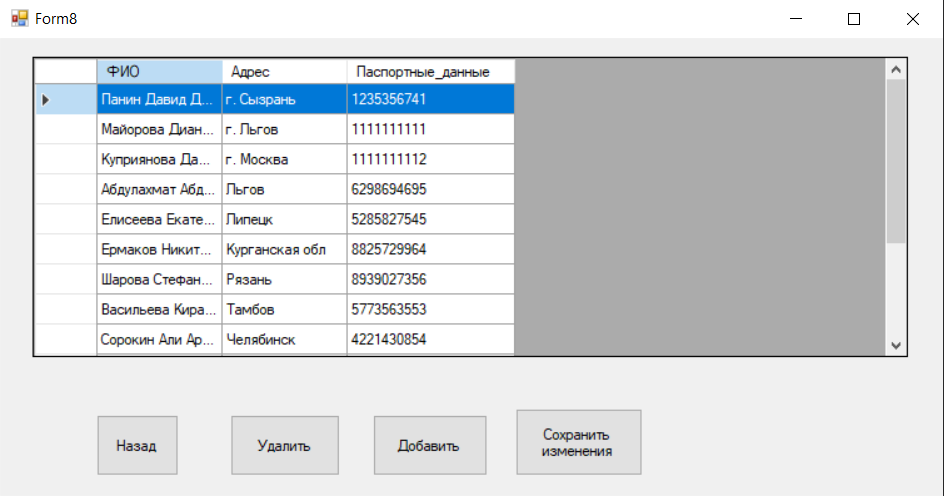


Рисунок 32– Форма добавления нового покупателя

После нажатия кнопки «Сотрудники», в главной форме, откроется окно с данными по персоналу автосалона.

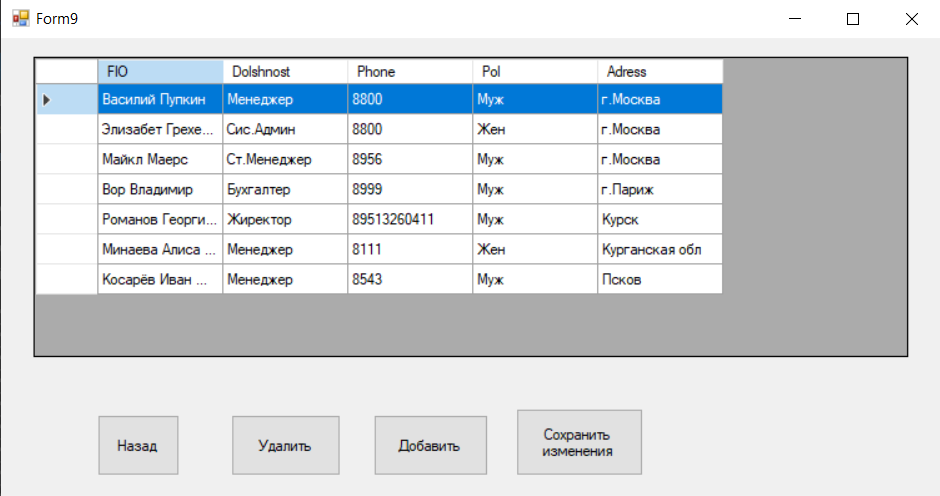


Рисунок 33– Форма добавления нового сотрудника

В клиентском приложении так же есть 3 специализированные выборки:

1. Выводит данные о машинах, которые привёз перегонщик за выбранный интервал времени.
2. Выводит данные о 5 самых лучших покупателей, которые принесли наибольшую прибыль автосалону за последний год.
3. Выводит данные о машинах от конкретного производителя за последний год.

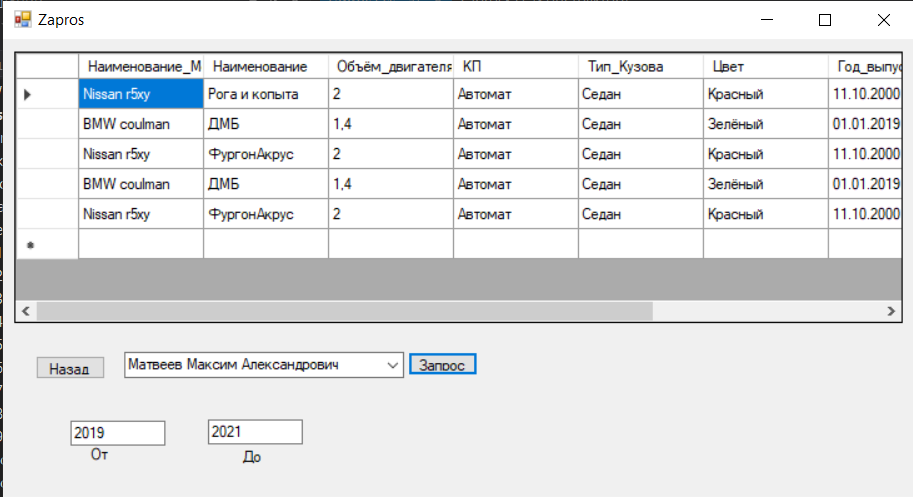


Рисунок 34– Форма выборки №1

SQL-запрос: "SELECT Модель\_авто, Наименование\_Модели, Поставщик, Наименование, Объём\_двигателя, КП, Тип\_Кузова, Цвет, Год\_выпуска, Код\_покупатель, ФИО, Адрес\_покупателя FROM autosalon.dbo.Автомобили JOIN Модель ON Код\_модели = Модель\_авто JOIN Поставщики ON Код\_поставщика=Поставщик JOIN Заказы ON Автомобиль = Код JOIN Покупатели ON Код\_покупателя = Код\_покупатель" + "WHERE Перегонщик = '" + comboBox1.SelectedValue + "'AND Дата\_поставки BETWEEN'" + textBox1.Text + "' AND '" + textBox2.Text + "'".

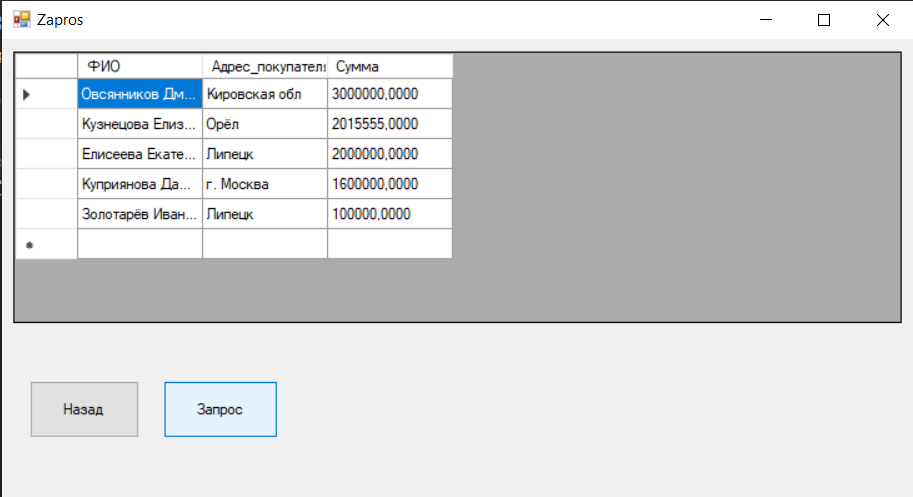


Рисунок 35– Форма выборки №2

SQL-запрос: "SELECT TOP (5) ФИО, Адрес\_покупателя, SUM(Цена) AS 'Сумма' FROM Покупатели JOIN Заказы ON Код\_покупатель = Код\_покупателя JOIN Автомобили ON Автомобиль = Код GROUP BY фИО, Адрес\_покупателя ORDER BY Сумма DESC";

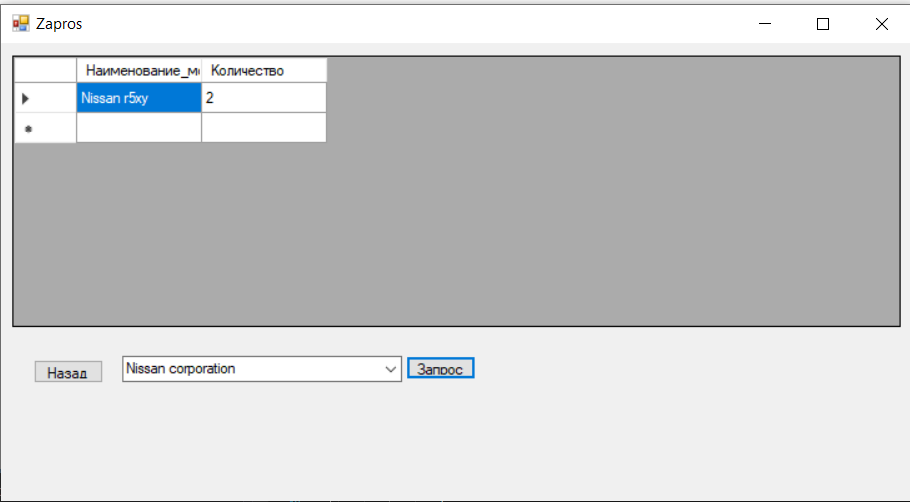


Рисунок 36– Форма выборки №3

SQL-запрос: "SELECT Наименование\_модели, COUNT(Модель\_авто) AS 'Количество' FROM Модель JOIN автомобили ON Модель\_авто = Код\_модели WHERE Дата\_поставки LIKE '%2020%' AND Код\_производителя ='" + comboBox1.SelectedValue + "' GROUP BY Наименование\_Модели";

Прикладная программа была протестирована, она работает без ошибок. Исходный код прикладной программы. (См. Листинг программы)

# **Заключение**

Данная работа выполнена на языке высокого уровня программирования в среде Visual studio 2019. При написании и разработке программного обеспечения использовались различные возможности C#, Windows Form и SQL.

Разработанная прикладная программа имеет удобный пользовательский интерфейс. Все элементы прикладной программы представлены как единый проект, управление которым осуществляется через окно главной формы проекта. Программа работает без ошибок, выполняет все функции.

В ходе работы курсовой работы были пройдены все основные этапы проектирования базы данных, составлена реляционная модель, выполнено рабочее проектирование схемы базы данных и интерфейса конечного пользователя. Были учтены особенности предметной области и составлены спецификации сущностей, атрибутов, связей, таблиц. В ходе работы проектирования интерфейса были составлены основные формы.

В результате выполнения курсовой работы были получены основные навыки анализа заданной предметной области, разработки баз данных, работы с СУБД Microsoft SQL Server и средой программирования.

Созданная база данных позволяет осуществлять ввод новых данных, изменять имеющиеся данные в базе и удалять старые. Данная программа очень удобна в использовании для систематизации данных.

**Список использованных источников**

1. Прайс, Дж. Oracle Database 11g. SQL: Операторы SQL и программы PL/SQL [Текст] /Дж. Прайс. - М.: ЛОРИ, 2018. - 660 c.
2. Черпаков, И.В. Основы программирования [Текст]: Учебник и практикум для прикладного бакалавриата/ И.В. Черпаков. - Люберцы: Юрайт, 2016. - 219 c.
3. Дейт, К. Дж. SQL и реляционная теория. Как грамотно писать код на SQL [Текст] / Дж. К. Дейт. - М.: Символ-плюс, 2016. - 469 c.
4. Дунаев, В.В. Базы данных. Язык SQL для студента [Текст] / В.В. Дунаев. - М.: БХВ-Петербург, 2017. - 288 c.
5. Карвин, Б. Программирование баз данных SQL. Типичные ошибки и их устранение [Текст] / Б. Карвин. - М.: Рид Групп, 2018. - 336 c.

# **Листинг программы**

Form1:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

Close();

}

private void button2\_Click(object sender, EventArgs e)

{

Form2 f2 = new Form2();

f2.Show();

}

private void button3\_Click(object sender, EventArgs e)

{

Form3 f3 = new Form3();

f3.Show();

}

private void button4\_Click(object sender, EventArgs e)

{

Form4 f4 = new Form4();

f4.Show();

}

private void button5\_Click(object sender, EventArgs e)

{

Form5 f5 = new Form5();

f5.Show();

}

private void button6\_Click(object sender, EventArgs e)

{

Form6 f6 = new Form6();

f6.Show();

}

private void button7\_Click(object sender, EventArgs e)

{

Form7 f7 = new Form7();

f7.Show();

}

private void button8\_Click(object sender, EventArgs e)

{

Form8 f8 = new Form8();

f8.Show();

}

private void button9\_Click(object sender, EventArgs e)

{

Form9 f9 = new Form9();

f9.Show();

}

private void button10\_Click(object sender, EventArgs e)

{

Connector.connectionString = "Data Source=" + textBox1.Text + ";Initial Catalog = autosalon; Integrated Security = True";

}

private void button11\_Click(object sender, EventArgs e)

{

Zapros1 zapros1 = new Zapros1();

zapros1.Show();

}

private void button12\_Click(object sender, EventArgs e)

{

Zapros2 zapros2 = new Zapros2();

zapros2.Show();

}

private void button13\_Click(object sender, EventArgs e)

{

Zapros3 zapros3 = new Zapros3();

zapros3.Show();

}

}

}

Form:2

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class Form2 : Form

{

DataSet ds;

SqlDataAdapter adapter;

string sql = "SELECT \* FROM Производители";

SqlCommandBuilder commandBuilder;

public Form2()

{

InitializeComponent();

dataGridView1.SelectionMode = DataGridViewSelectionMode.FullRowSelect;

dataGridView1.AllowUserToAddRows = false;

using (SqlConnection connection = new SqlConnection(Connector.connectionString))

{

connection.Open();

adapter = new SqlDataAdapter(sql, connection);

ds = new DataSet();

adapter.Fill(ds);

dataGridView1.DataSource = ds.Tables[0];

dataGridView1.Columns["Кодпроизводителя"].Visible = false;

}

}

private void Form2\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

Close();

}

private void button2\_Click(object sender, EventArgs e)

{

foreach (DataGridViewRow row in dataGridView1.SelectedRows)

{

dataGridView1.Rows.Remove(row);

}

}

private void button4\_Click(object sender, EventArgs e)

{

DataRow row = ds.Tables[0].NewRow();

ds.Tables[0].Rows.Add(row);

}

private void button5\_Click(object sender, EventArgs e)

{

try

{

using (SqlConnection connection = new SqlConnection(Connector.connectionString))

{

connection.Open();

adapter = new SqlDataAdapter(sql, connection);

commandBuilder = new SqlCommandBuilder(adapter);

adapter.Update(ds);

}

MessageBox.Show("Данные успешно обновлены");

}

catch

{

MessageBox.Show("Ошибка ввода данных");

}

}

}

}

Form3:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class Form3 : Form

{

DataSet ds;

SqlDataAdapter adapter;

string sql = "SELECT Код\_Модели, Марка, Тип\_Кузова, КП, Наименование\_Модели, Год\_выпуска, Цвет, Код\_Производителя, Название FROM Модель JOIN Производители ON Код\_производителя = Кодпроизводителя";

public Form3()

{

InitializeComponent();

dataGridView1.SelectionMode = DataGridViewSelectionMode.FullRowSelect;

dataGridView1.AllowUserToAddRows = false;

using (SqlConnection connection = new SqlConnection(Connector.connectionString))

{

connection.Open();

adapter = new SqlDataAdapter(sql, connection);

ds = new DataSet();

adapter.Fill(ds);

dataGridView1.DataSource = ds.Tables[0];

dataGridView1.Columns["Код\_Модели"].Visible = false;

dataGridView1.Columns["Код\_Производителя"].Visible = false;

dataGridView1.Columns["Название"].HeaderText = "Название производителя";

}

}

private void button1\_Click(object sender, EventArgs e)

{

Close();

}

private void button4\_Click(object sender, EventArgs e)

{

HelpFormModel f1 = new HelpFormModel();

f1.Show();

Close();

}

private void button2\_Click(object sender, EventArgs e)

{

foreach (DataGridViewRow row in dataGridView1.SelectedRows)

{

using (SqlConnection connection = new SqlConnection(Connector.connectionString))

{

SqlCommand command = new SqlCommand("DELETE FROM Модель WHERE Код=@kod", connection);

connection.Open();

SqlParameter Param1 = new SqlParameter("@kod", row.Cells[0].Value);

command.Parameters.Add(Param1);

try

{

command.ExecuteNonQuery();

MessageBox.Show("Удаление успешно");

}

catch (SqlException)

{

MessageBox.Show("Возникли ошибки");

}

connection.Close();

}

dataGridView1.Rows.Remove(row);

}

}

private void button6\_Click(object sender, EventArgs e)

{

for (int i = 0; i < dataGridView1.Rows.GetRowCount(DataGridViewElementStates.Selected); i++)

{

HelpFormModelupdate f1 = new HelpFormModelupdate();

f1.updatedata(dataGridView1.SelectedRows[i].Cells[0].Value.ToString(), dataGridView1.SelectedRows[i].Cells[1].Value.ToString(), dataGridView1.SelectedRows[i].Cells[2].Value.ToString(), dataGridView1.SelectedRows[i].Cells[3].Value.ToString(), dataGridView1.SelectedRows[i].Cells[4].Value.ToString(), dataGridView1.SelectedRows[i].Cells[5].Value.ToString(), dataGridView1.SelectedRows[i].Cells[6].Value.ToString(), dataGridView1.SelectedRows[i].Cells[7].Value.ToString(), dataGridView1.SelectedRows[i].Cells[8].Value.ToString());

f1.Show();

Close();

}

}

}

}

Form:4

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class Form4 : Form

{

DataSet ds;

SqlDataAdapter adapter;

string sql = "SELECT Код, Дата\_поставки,Модель\_авто,Наименование\_Модели, Цена, Объём\_двигателя, Номер\_двигателя, Перегонщик, ФИО, Поставщик, Наименование, Продано FROM Автомобили JOIN Модель ON Код\_Модели = Модель\_авто JOIN Перегонщики ON Код\_перегонщика = Перегонщик JOIN Поставщики ON Код\_поставщика = Поставщик";

public Form4()

{

InitializeComponent();

dataGridView1.SelectionMode = DataGridViewSelectionMode.FullRowSelect;

dataGridView1.AllowUserToAddRows = false;

using (SqlConnection connection = new SqlConnection(Connector.connectionString))

{

connection.Open();

adapter = new SqlDataAdapter(sql, connection);

ds = new DataSet();

adapter.Fill(ds);

dataGridView1.DataSource = ds.Tables[0];

dataGridView1.Columns["Код"].Visible = false;

dataGridView1.Columns["Модель\_авто"].Visible = false;

dataGridView1.Columns["Перегонщик"].Visible = false;

dataGridView1.Columns["Поставщик"].Visible = false;

dataGridView1.Columns["ФИО"].HeaderText = "ФИО перегонщика";

dataGridView1.Columns["Наименование"].HeaderText = "Наименование поставщика";

}

}

private void button1\_Click(object sender, EventArgs e)

{

Close();

}

private void button4\_Click(object sender, EventArgs e)

{

HelpFormAuto helpFormAuto = new HelpFormAuto();

helpFormAuto.Show();

}

private void button2\_Click(object sender, EventArgs e)

{

foreach (DataGridViewRow row in dataGridView1.SelectedRows)

{

using (SqlConnection connection = new SqlConnection(Connector.connectionString))

{

SqlCommand command = new SqlCommand("DELETE FROM Автомобили WHERE Код=@kod", connection);

connection.Open();

SqlParameter Param1 = new SqlParameter("@kod", row.Cells[0].Value);

command.Parameters.Add(Param1);

try

{

command.ExecuteNonQuery();

MessageBox.Show("Удаление успешно");

}

catch (SqlException)

{

MessageBox.Show("Возникли ошибки");

}

connection.Close();

}

dataGridView1.Rows.Remove(row);

}

}

private void button8\_Click(object sender, EventArgs e)

{

for (int i = 0; i < dataGridView1.Rows.GetRowCount(DataGridViewElementStates.Selected); i++)

{

HelpFormAuto\_update formAuto\_Update = new HelpFormAuto\_update();

formAuto\_Update.updatedataauto(dataGridView1.SelectedRows[i].Cells[0].Value.ToString(), dataGridView1.SelectedRows[i].Cells[1].Value.ToString(), dataGridView1.SelectedRows[i].Cells[2].Value.ToString(), dataGridView1.SelectedRows[i].Cells[3].Value.ToString(), dataGridView1.SelectedRows[i].Cells[4].Value.ToString(), dataGridView1.SelectedRows[i].Cells[5].Value.ToString(), dataGridView1.SelectedRows[i].Cells[6].Value.ToString(), dataGridView1.SelectedRows[i].Cells[7].Value.ToString(), dataGridView1.SelectedRows[i].Cells[8].Value.ToString(), dataGridView1.SelectedRows[i].Cells[9].Value.ToString(), dataGridView1.SelectedRows[i].Cells[10].Value.ToString(), dataGridView1.SelectedRows[i].Cells[11].Value);

formAuto\_Update.Show();

}

}

}

}

Form:5

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class Form5 : Form

{

DataSet ds;

SqlDataAdapter adapter;

string sql = "SELECT \* FROM Поставщики";

SqlCommandBuilder commandBuilder;

public Form5()

{

InitializeComponent();

dataGridView1.SelectionMode = DataGridViewSelectionMode.FullRowSelect;

dataGridView1.AllowUserToAddRows = false;

using (SqlConnection connection = new SqlConnection(Connector.connectionString))

{

connection.Open();

adapter = new SqlDataAdapter(sql, connection);

ds = new DataSet();

adapter.Fill(ds);

dataGridView1.DataSource = ds.Tables[0];

dataGridView1.Columns["Код\_поставщика"].Visible = false;

}

}

private void button1\_Click(object sender, EventArgs e)

{

Close();

}

private void button4\_Click(object sender, EventArgs e)

{

DataRow row = ds.Tables[0].NewRow();

ds.Tables[0].Rows.Add(row);

}

private void button2\_Click(object sender, EventArgs e)

{

foreach (DataGridViewRow row in dataGridView1.SelectedRows)

{

dataGridView1.Rows.Remove(row);

}

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void button5\_Click(object sender, EventArgs e)

{

try

{

using (SqlConnection connection = new SqlConnection(Connector.connectionString))

{

connection.Open();

adapter = new SqlDataAdapter(sql, connection);

commandBuilder = new SqlCommandBuilder(adapter);

adapter.Update(ds);

}

MessageBox.Show("Данные успешно обновлены");

}

catch

{

MessageBox.Show("Код поставщика не должен повторятся");

}

}

}

}

Form:6

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class Form6 : Form

{

DataSet ds;

SqlDataAdapter adapter;

string sql = "SELECT \* FROM Перегонщики";

SqlCommandBuilder commandBuilder;

public Form6()

{

InitializeComponent();

dataGridView1.SelectionMode = DataGridViewSelectionMode.FullRowSelect;

dataGridView1.AllowUserToAddRows = false;

using (SqlConnection connection = new SqlConnection(Connector.connectionString))

{

connection.Open();

adapter = new SqlDataAdapter(sql, connection);

ds = new DataSet();

adapter.Fill(ds);

dataGridView1.DataSource = ds.Tables[0];

dataGridView1.Columns["Код\_перегонщика"].Visible = false;

}

}

private void button1\_Click(object sender, EventArgs e)

{

Close();

}

private void button4\_Click(object sender, EventArgs e)

{

DataRow row = ds.Tables[0].NewRow();

ds.Tables[0].Rows.Add(row);

}

private void button2\_Click(object sender, EventArgs e)

{

foreach (DataGridViewRow row in dataGridView1.SelectedRows)

{

dataGridView1.Rows.Remove(row);

}

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void button5\_Click(object sender, EventArgs e)

{

try

{

using (SqlConnection connection = new SqlConnection(Connector.connectionString))

{

connection.Open();

adapter = new SqlDataAdapter(sql, connection);

commandBuilder = new SqlCommandBuilder(adapter);

adapter.Update(ds);

}

MessageBox.Show("Данные успешно обновлены");

}

catch

{

MessageBox.Show("Ошибка в заполнении данных");

}

}

}

}

Form:7

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class Form7 : Form

{

DataSet ds;

SqlDataAdapter adapter;

string sql = "SELECT Номер\_заказа, Дата\_заказа,Код\_покупателя, ФИО, Автомобиль, Номер\_двигателя, Цена, Статус\_оплаты, FIO FROM Заказы JOIN Покупатели ON Код\_покупателя = Код\_покупатель JOIN Personal ON Code\_personal = Продавец JOIN Автомобили ON Код = Автомобиль ";

public Form7()

{

InitializeComponent();

dataGridView1.SelectionMode = DataGridViewSelectionMode.FullRowSelect;

dataGridView1.AllowUserToAddRows = false;

using (SqlConnection connection = new SqlConnection(Connector.connectionString))

{

connection.Open();

adapter = new SqlDataAdapter(sql, connection);

ds = new DataSet();

adapter.Fill(ds);

dataGridView1.DataSource = ds.Tables[0];

dataGridView1.Columns["Код\_покупателя"].Visible = false;

dataGridView1.Columns["Автомобиль"].Visible = false;

}

}

private void button1\_Click(object sender, EventArgs e)

{

Close();

}

private void button4\_Click(object sender, EventArgs e)

{

HelpFormZakaz helpFormZakaz = new HelpFormZakaz();

helpFormZakaz.Show();

}

private void button2\_Click(object sender, EventArgs e)

{

foreach (DataGridViewRow row in dataGridView1.SelectedRows)

{

using (SqlConnection connection = new SqlConnection(Connector.connectionString))

{

SqlCommand command = new SqlCommand("DELETE FROM Заказы WHERE Номер\_заказа=@kod", connection);

connection.Open();

SqlParameter Param1 = new SqlParameter("@kod", row.Cells[0].Value);

command.Parameters.Add(Param1);

try

{

command.ExecuteNonQuery();

MessageBox.Show("Удаление успешно");

}

catch (SqlException)

{

MessageBox.Show("Возникли ошибки");

}

connection.Close();

}

dataGridView1.Rows.Remove(row);

}

}

private void button3\_Click(object sender, EventArgs e)

{

HelpFormPokypatel f1 = new HelpFormPokypatel();

f1.Show();

}

private void button3\_Click\_1(object sender, EventArgs e)

{

for (int i = 0; i < dataGridView1.Rows.GetRowCount(DataGridViewElementStates.Selected); i++)

{

HelpFormZakaz\_update helpFormZakaz\_Update = new HelpFormZakaz\_update();

helpFormZakaz\_Update.updatedatazakaz(dataGridView1.SelectedRows[i].Cells[0].Value.ToString(), dataGridView1.SelectedRows[i].Cells[1].Value.ToString(), dataGridView1.SelectedRows[i].Cells[3].Value.ToString(), dataGridView1.SelectedRows[i].Cells[5].Value.ToString(), dataGridView1.SelectedRows[i].Cells[6].Value.ToString(), dataGridView1.SelectedRows[i].Cells[7].Value.ToString(), dataGridView1.SelectedRows[i].Cells[8].Value.ToString());

helpFormZakaz\_Update.Show();

}

}

}

}

Form:8

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class Form8 : Form

{

DataSet ds;

SqlDataAdapter adapter;

string sql = "SELECT \* FROM Покупатели";

SqlCommandBuilder commandBuilder;

public Form8()

{

InitializeComponent();

dataGridView1.SelectionMode = DataGridViewSelectionMode.FullRowSelect;

dataGridView1.AllowUserToAddRows = false;

using (SqlConnection connection = new SqlConnection(Connector.connectionString))

{

connection.Open();

adapter = new SqlDataAdapter(sql, connection);

ds = new DataSet();

adapter.Fill(ds);

dataGridView1.DataSource = ds.Tables[0];

dataGridView1.Columns["Код\_покупателя"].Visible = false;

}

}

private void button1\_Click(object sender, EventArgs e)

{

Close();

}

private void button4\_Click(object sender, EventArgs e)

{

DataRow row = ds.Tables[0].NewRow();

ds.Tables[0].Rows.Add(row);

}

private void button2\_Click(object sender, EventArgs e)

{

foreach (DataGridViewRow row in dataGridView1.SelectedRows)

{

dataGridView1.Rows.Remove(row);

}

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void button5\_Click(object sender, EventArgs e)

{

try

{

using (SqlConnection connection = new SqlConnection(Connector.connectionString))

{

connection.Open();

adapter = new SqlDataAdapter(sql, connection);

commandBuilder = new SqlCommandBuilder(adapter);

adapter.Update(ds);

}

MessageBox.Show("Данные успешно обновлены");

}

catch

{

MessageBox.Show("Ошибка в заполнении данных");

}

}

}

}

Form:9

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class Form9 : Form

{

DataSet ds;

SqlDataAdapter adapter;

string sql = "SELECT \* FROM Personal";

SqlCommandBuilder commandBuilder;

public Form9()

{

InitializeComponent();

dataGridView1.SelectionMode = DataGridViewSelectionMode.FullRowSelect;

dataGridView1.AllowUserToAddRows = false;

using (SqlConnection connection = new SqlConnection(Connector.connectionString))

{

connection.Open();

adapter = new SqlDataAdapter(sql, connection);

ds = new DataSet();

adapter.Fill(ds);

dataGridView1.DataSource = ds.Tables[0];

dataGridView1.Columns["Code\_personal"].Visible = false;

}

}

private void button1\_Click(object sender, EventArgs e)

{

Close();

}

private void button4\_Click(object sender, EventArgs e)

{

DataRow row = ds.Tables[0].NewRow();

ds.Tables[0].Rows.Add(row);

}

private void button2\_Click(object sender, EventArgs e)

{

foreach (DataGridViewRow row in dataGridView1.SelectedRows)

{

dataGridView1.Rows.Remove(row);

}

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void button5\_Click(object sender, EventArgs e)

{

try

{

using (SqlConnection connection = new SqlConnection(Connector.connectionString))

{

connection.Open();

adapter = new SqlDataAdapter(sql, connection);

commandBuilder = new SqlCommandBuilder(adapter);

adapter.Update(ds);

}

MessageBox.Show("Данные успешно обновлены");

}

catch

{

MessageBox.Show("Ошибка в заполнении данных");

}

}

private void button3\_Click(object sender, EventArgs e)

{

HelpFormPersonal helpFormPersonal = new HelpFormPersonal();

helpFormPersonal.Show();

}

}

}

Form: HelpFormAuto-update

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class HelpFormAuto\_update : Form

{

DataSet ds;

SqlDataAdapter adapter;

string sql = "SELECT Код\_перегонщика,ФИО FROM Перегонщики; SELECT Код\_поставщика, Наименование FROM Поставщики; SELECT Код\_Модели, Наименование\_Модели FROM Модель";

string id\_auto;

string modelcode;

string peregonshik;

string postavshik;

string name\_postavshik;

string name\_peregon;

string name\_model;

public void updatedataauto(string idauto, string textbox1, string codemodel, string combobox3, string textbox2, string textbox3, string textbox4,string peregon, string combobox1,string postavsh, string combobox2, object checkbox1)

{

id\_auto = idauto;

textBox1.Text = textbox1;

modelcode = codemodel;

name\_model = combobox3;

textBox2.Text = textbox2;

textBox3.Text = textbox3;

textBox4.Text = textbox4;

peregonshik = peregon;

name\_peregon = combobox1;

postavshik = postavsh;

name\_postavshik = combobox2;

checkBox1.Checked = Convert.ToBoolean(checkbox1);

}

public HelpFormAuto\_update()

{

InitializeComponent();

}

private void HelpFormAuto\_update\_Load(object sender, EventArgs e)

{

using (SqlConnection connection = new SqlConnection(Connector.connectionString))

{

connection.Open();

adapter = new SqlDataAdapter(sql, connection);

ds = new DataSet();

adapter.Fill(ds);

comboBox1.DataSource = ds.Tables[0];

comboBox1.DisplayMember = "ФИО";

comboBox1.ValueMember = "Код\_перегонщика";

comboBox1.Text = name\_peregon;

comboBox2.DataSource = ds.Tables[1];

comboBox2.DisplayMember = "Наименование";

comboBox2.ValueMember = "Код\_поставщика";

comboBox2.Text = name\_postavshik;

comboBox3.DataSource = ds.Tables[2];

comboBox3.DisplayMember = "Наименование\_Модели";

comboBox3.ValueMember = "Код\_Модели";

comboBox3.Text = name\_model;

connection.Close();

}

}

private void button2\_Click(object sender, EventArgs e)

{

using (SqlConnection con = new SqlConnection(Connector.connectionString))

{

SqlCommand command = new SqlCommand("UPDATE Автомобили SET Дата\_поставки = @textbox1, Модель\_авто = @model\_auto, Цена = @textbox2, Объём\_двигателя = @textbox3, Номер\_двигателя = @textbox4, Перегонщик = @peregonshik, Поставщик = @postavshik, Продано = @checkbox1 WHERE Код = @id\_auto", con);

con.Open();

SqlParameter Param1 = new SqlParameter("@textbox1", textBox1.Text); //Дата\_поставки

SqlParameter Param2 = new SqlParameter("@model\_auto", comboBox3.SelectedValue); // модель\_авто

SqlParameter Param4 = new SqlParameter("@textbox2", Convert.ToSingle(textBox2.Text)); //Цена

SqlParameter Param5 = new SqlParameter("@textbox3", textBox3.Text); // Объём двигателя

SqlParameter Param6 = new SqlParameter("@textbox4", textBox4.Text); // номер двигателя

SqlParameter Param7 = new SqlParameter("@peregonshik", comboBox1.SelectedValue); // перегонщик

SqlParameter Param9 = new SqlParameter("@postavshik", comboBox2.SelectedValue); // поставщик

SqlParameter Param11 = new SqlParameter("@checkbox1", checkBox1.Checked); // наименование поставщика

SqlParameter Param12 = new SqlParameter("@id\_auto", id\_auto); // номер авто

command.Parameters.Add(Param1);

command.Parameters.Add(Param2);

command.Parameters.Add(Param4);

command.Parameters.Add(Param5);

command.Parameters.Add(Param6);

command.Parameters.Add(Param7);

command.Parameters.Add(Param9);

command.Parameters.Add(Param11);

command.Parameters.Add(Param12);

try

{

command.ExecuteNonQuery();

MessageBox.Show("Обновление успешно");

}

catch (SqlException)

{

MessageBox.Show("Возникли ошибки");

}

con.Close();

}

}

private void button1\_Click(object sender, EventArgs e)

{

Close();

}

/\*

private void button3\_Click(object sender, EventArgs e)

{

HelpFormModel f1 = new HelpFormModel();

f1.Show();

}

private void button6\_Click(object sender, EventArgs e)

{

HelpFormPeregonshiki f2 = new HelpFormPeregonshiki();

f2.Show();

}

private void button7\_Click(object sender, EventArgs e)

{

HelpFormPostavshick f3 = new HelpFormPostavshick();

f3.Show();

}

\*/

}

}

Form: HelpFormAuto

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class HelpFormAuto : Form

{

DataSet ds;

SqlDataAdapter adapter;

string sql = "SELECT Код\_перегонщика,ФИО FROM Перегонщики; SELECT Код\_поставщика, Наименование FROM Поставщики; SELECT Код\_Модели, Наименование\_Модели FROM Модель";

public HelpFormAuto()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

Close();

}

private void button2\_Click(object sender, EventArgs e)

{

using (SqlConnection con = new SqlConnection(Connector.connectionString))

{

SqlCommand command = new SqlCommand("INSERT INTO autosalon.dbo.Автомобили (Дата\_поставки, Модель\_авто, Цена, Объём\_двигателя, Номер\_двигателя, Перегонщик, Поставщик, Продано) VALUES (@textBox1, @model\_auto, @textbox2,@textbox3,@textbox4, @peregonshik, @postavshik, @checkbox1)", con);

con.Open();

SqlParameter Param1 = new SqlParameter("@textbox1", textBox1.Text); //Дата\_поставки

SqlParameter Param2 = new SqlParameter("@model\_auto", comboBox3.SelectedValue); // модель\_авто

SqlParameter Param4 = new SqlParameter("@textbox2", textBox2.Text); //Цена

SqlParameter Param5 = new SqlParameter("@textbox3", textBox3.Text); // Объём двигателя

SqlParameter Param6 = new SqlParameter("@textbox4", textBox4.Text); // номер двигателя

SqlParameter Param7 = new SqlParameter("@peregonshik", comboBox1.SelectedValue); // перегонщик

SqlParameter Param9 = new SqlParameter("@postavshik", comboBox2.SelectedValue); // поставщик

SqlParameter Param11 = new SqlParameter("@checkbox1", checkBox1.Checked); // наименование поставщика

command.Parameters.Add(Param1);

command.Parameters.Add(Param2);

command.Parameters.Add(Param4);

command.Parameters.Add(Param5);

command.Parameters.Add(Param6);

command.Parameters.Add(Param7);

command.Parameters.Add(Param9);

command.Parameters.Add(Param11);

try

{

command.ExecuteNonQuery();

MessageBox.Show("Добавление успешно");

}

catch (SqlException)

{

MessageBox.Show("Возникли ошибки");

}

con.Close();

}

}

private void HelpFormAuto\_Load(object sender, EventArgs e)

{

using (SqlConnection connection = new SqlConnection(Connector.connectionString))

{

connection.Open();

adapter = new SqlDataAdapter(sql, connection);

ds = new DataSet();

adapter.Fill(ds);

comboBox1.DataSource = ds.Tables[0];

comboBox1.DisplayMember = "ФИО";

comboBox1.ValueMember = "Код\_перегонщика";

comboBox2.DataSource = ds.Tables[1];

comboBox2.DisplayMember = "Наименование";

comboBox2.ValueMember = "Код\_поставщика";

comboBox3.DataSource = ds.Tables[2];

comboBox3.DisplayMember = "Наименование\_Модели";

comboBox3.ValueMember = "Код\_Модели";

connection.Close();

}

}

private void button3\_Click(object sender, EventArgs e)

{

HelpFormModel f1 = new HelpFormModel();

f1.Show();

}

private void button6\_Click(object sender, EventArgs e)

{

HelpFormPeregonshiki f2 = new HelpFormPeregonshiki();

f2.Show();

}

private void button7\_Click(object sender, EventArgs e)

{

HelpFormPostavshick f3 = new HelpFormPostavshick();

f3.Show();

}

}

}

Form: HelpFormModel – update

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class HelpFormModelupdate : Form

{

DataSet ds;

SqlDataAdapter adapter;

string sql = "SELECT Кодпроизводителя, Название FROM Производители";

string proizvodbox;

string modelcode;

public void updatedata(string codemodel,string textbox1,string textbox2,string combobox1, string textbox3, string textbox4, string textbox5, string id, string combobox2)

{

modelcode = codemodel;

textBox1.Text = textbox1;

textBox2.Text = textbox2;

comboBox1.Text = combobox1;

textBox3.Text = textbox3;

textBox4.Text = textbox4;

textBox5.Text = textbox5;

proizvodbox = combobox2;

}

public HelpFormModelupdate()

{

InitializeComponent();

}

private void HelpFormModel\_Load(object sender, EventArgs e)

{

using (SqlConnection connection = new SqlConnection(Connector.connectionString))

{

connection.Open();

adapter = new SqlDataAdapter(sql, connection);

ds = new DataSet();

adapter.Fill(ds);

comboBox2.DataSource = ds.Tables[0];

comboBox2.DisplayMember = "Название";

comboBox2.ValueMember = "Кодпроизводителя";

comboBox2.Text = proizvodbox;

connection.Close();

}

}

private void button1\_Click(object sender, EventArgs e)

{

Close();

Form3 form3 = new Form3();

form3.Show();

}

private void button2\_Click(object sender, EventArgs e)

{

using (SqlConnection con = new SqlConnection(Connector.connectionString))

{

SqlCommand command = new SqlCommand("UPDATE Модель SET Марка = @textbox1, Тип\_Кузова = @textbox2 , КП = @kp, Наименование\_Модели = @textbox3, Год\_выпуска = @textbox4, Цвет = @textbox5, Код\_производителя = @id WHERE Код = @modelcode ", con);

con.Open();

SqlParameter Param1 = new SqlParameter("@textbox1", textBox1.Text);

SqlParameter Param2 = new SqlParameter("@textbox2", textBox2.Text);

SqlParameter Param3 = new SqlParameter("@textbox3", textBox3.Text);

SqlParameter Param4 = new SqlParameter("@kp", comboBox1.SelectedItem);

SqlParameter Param5 = new SqlParameter("@textbox4", textBox4.Text);

SqlParameter Param6 = new SqlParameter("@textbox5", textBox5.Text);

SqlParameter Param7 = new SqlParameter("@id", comboBox2.SelectedValue);

SqlParameter Param9 = new SqlParameter("@modelcode", modelcode);

command.Parameters.Add(Param1);

command.Parameters.Add(Param2);

command.Parameters.Add(Param3);

command.Parameters.Add(Param4);

command.Parameters.Add(Param5);

command.Parameters.Add(Param6);

command.Parameters.Add(Param7);

command.Parameters.Add(Param9);

try

{

command.ExecuteNonQuery();

MessageBox.Show("Обновление успешно");

}

catch (SqlException)

{

MessageBox.Show("Возникли ошибки");

}

command.ExecuteNonQuery();

con.Close();

}

}

/\*

private void button3\_Click(object sender, EventArgs e)

{

HelpFormProizvoditeli fh = new HelpFormProizvoditeli();

fh.Show();

Close();

}

\*/

}

}

Form: HelpFormModel

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class HelpFormModel : Form

{

DataSet ds;

SqlDataAdapter adapter;

string sql = "SELECT Кодпроизводителя, Название FROM Производители";

public HelpFormModel()

{

InitializeComponent();

}

private void HelpFormModel\_Load(object sender, EventArgs e)

{

comboBox1.Text = "Автомат";

using (SqlConnection connection = new SqlConnection(Connector.connectionString))

{

connection.Open();

adapter = new SqlDataAdapter(sql, connection);

ds = new DataSet();

adapter.Fill(ds);

comboBox2.DataSource = ds.Tables[0];

comboBox2.DisplayMember = "Название";

comboBox2.ValueMember = "Кодпроизводителя";

connection.Close();

}

}

private void button1\_Click(object sender, EventArgs e)

{

Close();

}

private void button2\_Click(object sender, EventArgs e)

{

using (SqlConnection con = new SqlConnection(Connector.connectionString))

{

SqlCommand command = new SqlCommand("INSERT INTO autosalon.dbo.Модель (Марка, Тип\_Кузова, КП, Наименование\_Модели, Год\_выпуска, Цвет, Код\_производителя) VALUES (@textBox1, @textBox2,@kp,@textBox3,@textBox4,@textBox5,@id)", con);

con.Open();

SqlParameter Param1 = new SqlParameter("@textbox1", textBox1.Text);

SqlParameter Param2 = new SqlParameter("@textbox2", textBox2.Text);

SqlParameter Param3 = new SqlParameter("@textbox3", textBox3.Text);

SqlParameter Param4 = new SqlParameter("@kp", comboBox1.SelectedItem);

SqlParameter Param5 = new SqlParameter("@textbox4", textBox4.Text);

SqlParameter Param6 = new SqlParameter("@textbox5", textBox5.Text);

SqlParameter Param7 = new SqlParameter("@id", comboBox2.SelectedValue);

command.Parameters.Add(Param1);

command.Parameters.Add(Param2);

command.Parameters.Add(Param3);

command.Parameters.Add(Param4);

command.Parameters.Add(Param5);

command.Parameters.Add(Param6);

command.Parameters.Add(Param7);

try

{

command.ExecuteNonQuery();

MessageBox.Show("Добавление успешно");

}

catch (SqlException)

{

MessageBox.Show("Возникли ошибки");

}

con.Close();

}

}

private void button3\_Click(object sender, EventArgs e)

{

HelpFormProizvoditeli fh = new HelpFormProizvoditeli();

fh.Show();

}

}

}

Form: HelpFormPeregonshiki

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class HelpFormPeregonshiki : Form

{

public HelpFormPeregonshiki()

{

InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(Connector.connectionString);

SqlCommand command = new SqlCommand("INSERT INTO autosalon.dbo.Поставщики (ФИО, Паспорт\_данные) VALUES (@textBox1, @textBox2)", con);

con.Open();

SqlParameter Param1 = new SqlParameter("@textbox1", textBox1.Text);

SqlParameter Param2 = new SqlParameter("@textbox2", textBox2.Text);

command.Parameters.Add(Param1);

command.Parameters.Add(Param2);

try

{

command.ExecuteNonQuery();

MessageBox.Show("Добавление успешно");

}

catch (SqlException)

{

MessageBox.Show("Возникли ошибки");

}

con.Close();

}

private void button1\_Click(object sender, EventArgs e)

{

Close();

}

}

}

Form: HelpFormPersonal

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class HelpFormPersonal : Form

{

public HelpFormPersonal()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

Close();

}

private void button2\_Click(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(Connector.connectionString);

SqlCommand command = new SqlCommand("INSERT INTO autosalon.dbo.Personal (FIO, Dolshnost, Phone, Pol, Adress) VALUES (@textBox1, @textBox2,@textBox3,@textBox4,@textBox5)", con);

con.Open();

SqlParameter Param1 = new SqlParameter("@textbox1", textBox1.Text);

SqlParameter Param2 = new SqlParameter("@textbox2", textBox2.Text);

SqlParameter Param3 = new SqlParameter("@textbox3", textBox3.Text);

SqlParameter Param4 = new SqlParameter("@textbox4", comboBox1.Text);

SqlParameter Param5 = new SqlParameter("@textbox5", textBox5.Text);

command.Parameters.Add(Param1);

command.Parameters.Add(Param2);

command.Parameters.Add(Param3);

command.Parameters.Add(Param4);

command.Parameters.Add(Param5);

try

{

command.ExecuteNonQuery();

MessageBox.Show("Добавление успешно");

}

catch (SqlException)

{

MessageBox.Show("Возникли ошибки");

}

con.Close();

}

}

}

Form: HelpFormPokypatel

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class HelpFormPokypatel : Form

{

public HelpFormPokypatel()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

Close();

}

private void button2\_Click(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(Connector.connectionString);

SqlCommand command = new SqlCommand("INSERT INTO autosalon.dbo.Покупатели (ФИО, Адрес, Паспортные\_данные) VALUES (@textBox1, @textBox2,@textBox3)", con);

con.Open();

SqlParameter Param1 = new SqlParameter("@textbox1", textBox1.Text);

SqlParameter Param2 = new SqlParameter("@textbox2", textBox2.Text);

SqlParameter Param3 = new SqlParameter("@textbox3", textBox3.Text);

command.Parameters.Add(Param1);

command.Parameters.Add(Param2);

command.Parameters.Add(Param3);

try

{

command.ExecuteNonQuery();

MessageBox.Show("Добавление успешно");

}

catch (SqlException)

{

MessageBox.Show("Возникли ошибки");

}

con.Close();

}

}

}

Form: HelpFormPokypatel

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class HelpFormPokypatel : Form

{

public HelpFormPokypatel()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

Close();

}

private void button2\_Click(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(Connector.connectionString);

SqlCommand command = new SqlCommand("INSERT INTO autosalon.dbo.Покупатели (ФИО, Адрес, Паспортные\_данные) VALUES (@textBox1, @textBox2,@textBox3)", con);

con.Open();

SqlParameter Param1 = new SqlParameter("@textbox1", textBox1.Text);

SqlParameter Param2 = new SqlParameter("@textbox2", textBox2.Text);

SqlParameter Param3 = new SqlParameter("@textbox3", textBox3.Text);

command.Parameters.Add(Param1);

command.Parameters.Add(Param2);

command.Parameters.Add(Param3);

try

{

command.ExecuteNonQuery();

MessageBox.Show("Добавление успешно");

}

catch (SqlException)

{

MessageBox.Show("Возникли ошибки");

}

con.Close();

}

}

}

Form: HelpFormPostavshick

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class HelpFormPostavshick : Form

{

public HelpFormPostavshick()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

Close();

}

private void button2\_Click(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(Connector.connectionString);

SqlCommand command = new SqlCommand("INSERT INTO autosalon.dbo.Поставщики (ИНН, Наименование,Реквизиты\_банка, Адрес) VALUES (@textBox1, @textBox2,@textBox3,@textBox4)", con);

con.Open();

SqlParameter Param1 = new SqlParameter("@textbox1", textBox1.Text);

SqlParameter Param2 = new SqlParameter("@textbox2", textBox2.Text);

SqlParameter Param3 = new SqlParameter("@textbox3", textBox3.Text);

SqlParameter Param4 = new SqlParameter("@textbox4", textBox4.Text);

command.Parameters.Add(Param1);

command.Parameters.Add(Param2);

command.Parameters.Add(Param3);

command.Parameters.Add(Param4);

try

{

command.ExecuteNonQuery();

MessageBox.Show("Добавление успешно");

}

catch (SqlException)

{

MessageBox.Show("Возникли ошибки");

}

con.Close();

}

}

}

Form: HelpFormProizvoditeli

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class HelpFormProizvoditeli : Form

{

public HelpFormProizvoditeli()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

Close();

HelpFormModel helpFormModel = new HelpFormModel();

helpFormModel.Show();

}

private void button2\_Click(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(Connector.connectionString);

SqlCommand command = new SqlCommand("INSERT INTO autosalon.dbo.Производители (Название, Страна) VALUES (@textBox1, @textBox2)",con);

con.Open();

SqlParameter Param1 = new SqlParameter("@textbox1", textBox1.Text);

SqlParameter Param2 = new SqlParameter("@textbox2", textBox2.Text);

command.Parameters.Add(Param1);

command.Parameters.Add(Param2);

try

{

command.ExecuteNonQuery();

MessageBox.Show("Добавление успешно");

}

catch (SqlException)

{

MessageBox.Show("Возникли ошибки");

}

con.Close();

}

}

}

Form: HelpFormZakaz-update

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class HelpFormZakaz\_update : Form

{

DataSet ds;

SqlDataAdapter adapter;

string sql = "SELECT Код\_покупателя, ФИО FROM Покупатели; SELECT Код, Номер\_двигателя, Цена, Продано FROM Автомобили; SELECT Code\_personal, FIO FROM Personal WHERE Dolshnost = 'Менеджер' OR Dolshnost = 'Ст.Менеджер'";

string id\_zakaz;

string combobof1;

string combobof2;

string combobof3;

string texbox2;

public void updatedatazakaz(string idzakaz, string textbox1, string combobox1, string combobox2, string textbox2, string checkbox, string combobox3)

{

id\_zakaz = idzakaz;

textBox1.Text = textbox1;

combobof1 = combobox1;

combobof2 = combobox2;

texbox2 = textbox2;

checkBox1.Checked = Convert.ToBoolean(checkbox);

combobof3 = combobox3;

}

public HelpFormZakaz\_update()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

Close();

}

private void HelpFormZakaz\_update\_Load(object sender, EventArgs e)

{

using (SqlConnection connection = new SqlConnection(Connector.connectionString))

{

connection.Open();

adapter = new SqlDataAdapter(sql, connection);

ds = new DataSet();

adapter.Fill(ds);

comboBox1.DataSource = ds.Tables[0];

comboBox1.DisplayMember = "ФИО";

comboBox1.ValueMember = "Код\_покупателя";

comboBox1.Text = combobof1;

comboBox2.DataSource = ds.Tables[1];

comboBox2.DisplayMember = "Номер\_двигателя";

comboBox2.ValueMember = "Код";

comboBox2.Text = combobof2;

textBox2.Text = ds.Tables[1].Rows[0][2].ToString();

textBox2.Text = texbox2;

comboBox3.DataSource = ds.Tables[2];

comboBox3.DisplayMember = "FIO";

comboBox3.ValueMember = "Code\_personal";

comboBox3.Text = combobof3;

connection.Close();

}

}

private void button2\_Click(object sender, EventArgs e)

{

using (SqlConnection con = new SqlConnection(Connector.connectionString))

{

SqlCommand command = new SqlCommand("UPDATE autosalon.dbo.Заказы SET Дата\_заказа = @textbox1, Код\_покупатель = @combobox1val, Автомобиль = @combobox2val, Статус\_оплаты = @checkbox, Продавец = @combobox3val WHERE Номер\_заказа = @idzakaz", con);

con.Open();

SqlParameter Param0 = new SqlParameter("@idzakaz", id\_zakaz); //id заказа

SqlParameter Param1 = new SqlParameter("@textbox1", textBox1.Text); //Дата заказа

SqlParameter Param2 = new SqlParameter("@combobox1val", comboBox1.SelectedValue); // Код покупателя

SqlParameter Param4 = new SqlParameter("@combobox2val", comboBox2.SelectedValue); // Номер авто

SqlParameter Param7 = new SqlParameter("@checkbox", checkBox1.Checked); // статус оплаты

SqlParameter Param8 = new SqlParameter("@combobox3val", comboBox3.SelectedValue); // номер продавца

command.Parameters.Add(Param0);

command.Parameters.Add(Param1);

command.Parameters.Add(Param2);

command.Parameters.Add(Param4);

command.Parameters.Add(Param7);

command.Parameters.Add(Param8);

try

{

command.ExecuteNonQuery();

MessageBox.Show("Добавление успешно");

}

catch (SqlException)

{

MessageBox.Show("Возникли ошибки");

}

con.Close();

}

}

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

textBox2.Text = ds.Tables[1].Rows[comboBox2.SelectedIndex][2].ToString();

}

}

}

Form: HelpFormZakaz

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class HelpFormZakaz : Form

{

DataSet ds;

SqlDataAdapter adapter;

string sql = "SELECT Код\_покупателя, ФИО FROM Покупатели; SELECT Код, Номер\_двигателя, Цена, Продано FROM Автомобили; SELECT Code\_personal, FIO FROM Personal WHERE Dolshnost = 'Менеджер' OR Dolshnost = 'Ст.Менеджер'";

public HelpFormZakaz()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

Close();

}

private void button2\_Click(object sender, EventArgs e)

{

using (SqlConnection con = new SqlConnection(Connector.connectionString))

{

SqlCommand command = new SqlCommand("INSERT INTO autosalon.dbo.Заказы (Дата\_заказа, Код\_покупатель, Автомобиль, Статус\_оплаты, Продавец) VALUES (@textbox1, @combobox1val, @combobox2val, @checkbox, @combobox3val)", con);

con.Open();

SqlParameter Param1 = new SqlParameter("@textbox1", textBox1.Text); //Дата заказа

SqlParameter Param2 = new SqlParameter("@combobox1val", comboBox1.SelectedValue); // Код покупателя

SqlParameter Param4 = new SqlParameter("@combobox2val", comboBox2.SelectedValue); // Номер авто

SqlParameter Param6 = new SqlParameter("@textbox2", textBox2.Text); // Сумма оплаты

SqlParameter Param7 = new SqlParameter("@checkbox", checkBox1.Checked); // статус оплаты

SqlParameter Param8 = new SqlParameter("@combobox3val", comboBox3.SelectedValue); // номер продавца

command.Parameters.Add(Param1);

command.Parameters.Add(Param2);

command.Parameters.Add(Param4);

command.Parameters.Add(Param6);

command.Parameters.Add(Param7);

command.Parameters.Add(Param8);

try

{

command.ExecuteNonQuery();

MessageBox.Show("Добавление успешно");

}

catch (SqlException)

{

MessageBox.Show("Возникли ошибки");

}

con.Close();

}

}

private void HelpFormZakaz\_Load(object sender, EventArgs e)

{

using (SqlConnection connection = new SqlConnection(Connector.connectionString))

{

connection.Open();

adapter = new SqlDataAdapter(sql, connection);

ds = new DataSet();

adapter.Fill(ds);

comboBox1.DataSource = ds.Tables[0];

comboBox1.DisplayMember = "ФИО";

comboBox1.ValueMember = "Код\_покупателя";

comboBox2.DataSource = ds.Tables[1];

comboBox2.DisplayMember = "Номер\_двигателя";

comboBox2.ValueMember = "Код";

textBox2.Text = ds.Tables[1].Rows[0][2].ToString();

comboBox3.DataSource = ds.Tables[2];

comboBox3.DisplayMember = "FIO";

comboBox3.ValueMember = "Code\_personal";

connection.Close();

}

}

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

textBox2.Text = ds.Tables[1].Rows[comboBox2.SelectedIndex][2].ToString();

}

private void button3\_Click(object sender, EventArgs e)

{

HelpFormPokypatel helpFormPokypatel = new HelpFormPokypatel();

helpFormPokypatel.Show();

}

private void button6\_Click(object sender, EventArgs e)

{

HelpFormAuto helpFormAuto = new HelpFormAuto();

helpFormAuto.Show();

}

private void button7\_Click(object sender, EventArgs e)

{

HelpFormPersonal helpFormPersonal = new HelpFormPersonal();

helpFormPersonal.Show();

}

}

}

Form: Zapros1

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class Zapros1 : Form

{

DataSet ds;

DataSet ds2;

SqlDataAdapter adapter;

string sql2 = "SELECT Код\_перегонщика, ФИО FROM Перегонщики";

public Zapros1()

{

InitializeComponent();

}

private void Zapros1\_Load(object sender, EventArgs e)

{

using (SqlConnection connection = new SqlConnection(Connector.connectionString))

{

connection.Open();

adapter = new SqlDataAdapter(sql2, connection);

ds2 = new DataSet();

adapter.Fill(ds2);

comboBox1.DataSource = ds2.Tables[0];

comboBox1.DisplayMember = "ФИО";

comboBox1.ValueMember = "Код\_перегонщика";

comboBox1.Text = "Выберите перегонщика";

connection.Close();

}

}

private void button2\_Click(object sender, EventArgs e)

{

Close();

}

private void button1\_Click(object sender, EventArgs e)

{

string sql = "SELECT Модель\_авто, Наименование\_Модели, Поставщик, Наименование, Объём\_двигателя, КП, Тип\_Кузова, Цвет, Год\_выпуска, Код\_покупатель, ФИО, Адрес\_покупателя FROM autosalon.dbo.Автомобили JOIN Модель ON Код\_модели = Модель\_авто JOIN Поставщики ON Код\_поставщика=Поставщик JOIN Заказы ON Автомобиль = Код JOIN Покупатели ON Код\_покупателя = Код\_покупатель" +

" WHERE Перегонщик = '" + comboBox1.SelectedValue + "'AND Дата\_поставки BETWEEN'" + textBox1.Text + "' AND '" + textBox2.Text + "'";

using (SqlConnection con = new SqlConnection(Connector.connectionString))

{

try

{

con.Open();

adapter = new SqlDataAdapter(sql, con);

ds = new DataSet();

ds.Clear();

adapter.Fill(ds);

dataGridView1.DataSource = ds.Tables[0];

con.Close();

}

catch (SqlException)

{

MessageBox.Show("Возникли ошибки");

}

dataGridView1.Columns["Модель\_авто"].Visible = false;

dataGridView1.Columns["Поставщик"].Visible = false;

dataGridView1.Columns["Код\_покупатель"].Visible = false;

}

}

}

}

Form: Zapros2

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class Zapros2 : Form

{

DataSet ds;

DataSet ds2;

SqlDataAdapter adapter;

string sql2 = "SELECT Кодпроизводителя, Название FROM Производители";

public Zapros2()

{

InitializeComponent();

}

private void Zapros2\_Load(object sender, EventArgs e)

{

using (SqlConnection connection = new SqlConnection(Connector.connectionString))

{

connection.Open();

adapter = new SqlDataAdapter(sql2, connection);

ds2 = new DataSet();

adapter.Fill(ds2);

comboBox1.DataSource = ds2.Tables[0];

comboBox1.DisplayMember = "Название";

comboBox1.ValueMember = "Кодпроизводителя";

comboBox1.Text = "Выберите производителя";

connection.Close();

}

}

private void button2\_Click(object sender, EventArgs e)

{

Close();

}

private void button1\_Click(object sender, EventArgs e)

{

string sql = "SELECT Наименование\_модели, COUNT(Модель\_авто) AS 'Количество' FROM Модель JOIN автомобили ON Модель\_авто = Код\_модели WHERE Дата\_поставки LIKE '%2020%' AND Код\_производителя ='" + comboBox1.SelectedValue + "' GROUP BY Наименование\_Модели";

using (SqlConnection con = new SqlConnection(Connector.connectionString))

{

try

{

con.Open();

adapter = new SqlDataAdapter(sql, con);

ds = new DataSet();

ds.Clear();

adapter.Fill(ds);

dataGridView1.DataSource = ds.Tables[0];

con.Close();

}

catch (SqlException)

{

MessageBox.Show("Возникли ошибки");

}

}

}

}

}

Form: Zapros3

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class Zapros3 : Form

{

DataSet ds;

DataSet ds2;

SqlDataAdapter adapter;

string sql2 = "SELECT Кодпроизводителя, Название FROM Производители";

public Zapros3()

{

InitializeComponent();

}

private void Zapros3\_Load(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

Close();

}

private void button1\_Click(object sender, EventArgs e)

{

string sql = "SELECT TOP (5) ФИО, Адрес\_покупателя, SUM(Цена) AS 'Сумма' FROM Покупатели JOIN Заказы ON Код\_покупатель = Код\_покупателя JOIN Автомобили ON Автомобиль = Код GROUP BY фИО, Адрес\_покупателя ORDER BY Сумма DESC";

using (SqlConnection con = new SqlConnection(Connector.connectionString))

{

try

{

con.Open();

adapter = new SqlDataAdapter(sql, con);

ds = new DataSet();

ds.Clear();

adapter.Fill(ds);

dataGridView1.DataSource = ds.Tables[0];

con.Close();

}

catch (SqlException)

{

MessageBox.Show("Возникли ошибки");

}

}

}

}

}

Класс: Connector

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace WindowsFormsApp1

{

class Connector

{

public static string connectionString = @"Data Source=DESKTOP-4OC7QLJ\SQLEXPRESS;Initial Catalog=autosalon;Integrated Security=True";

}

}