Для начала работы необходимо запустить приложение Microsoft SQL Server Management Studio (рис.1)



Рисунок 1 - Приложение SQL Server

После запуска приложения появится окно «Соединение с сервером» (рис.2), в котором нужно выбрать тип сервера, имя сервера (будет отличаться от примера) и проверку подлинности. Когда вы заполнили все поля, необходимо нажать кнопку «Соединить».

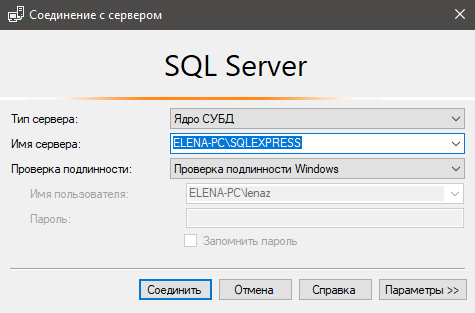


Рисунок 2 - Соединение с сервером

После нажатия кнопки «Соединить» откроется окно, в котором необходимо создать базу данных. Для этого нужно перейти к «Обозревателю объектов» (рис.3), который находится в левой части экрана.

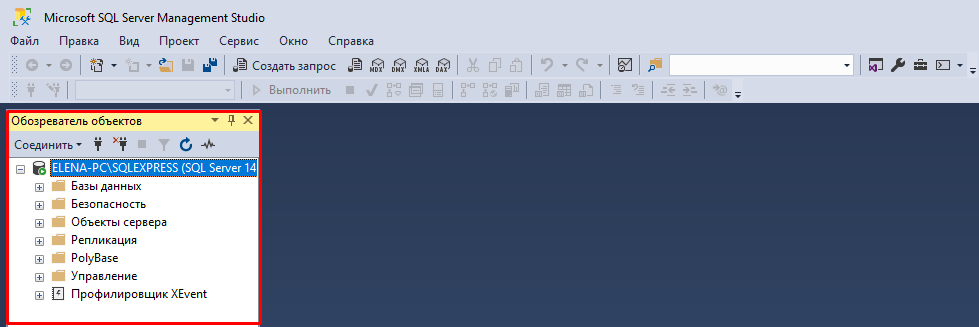


Рисунок 3 - Обозреватель объектов

В «Обозревателе объектов» нужно правой кнопкой мыши нажать на папку «Базы данных» и в появившемся списке выбрать «Создать базу данных…» (рис.4).

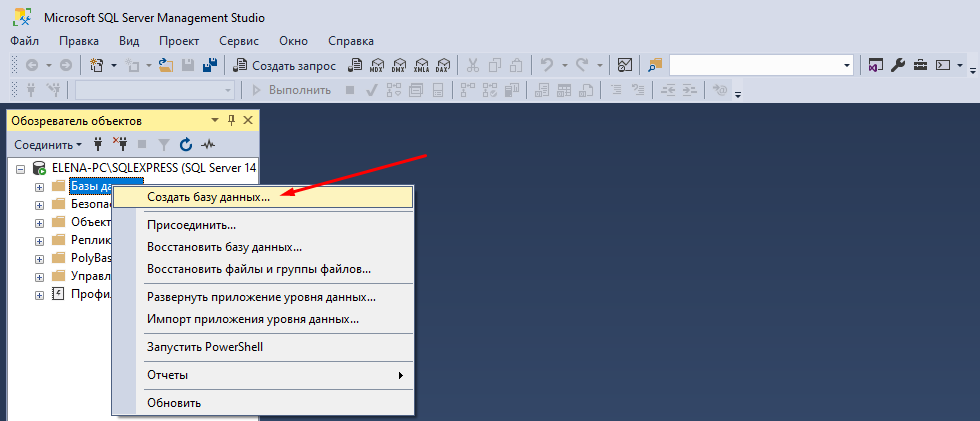


Рисунок 4 - Создание базы данных

В появившемся окне «Создание базы данных» (рис.5) вы должны ввести имя базы данных (Chernovik) в соответствующее поле, после чего нажать кнопку «ОК».

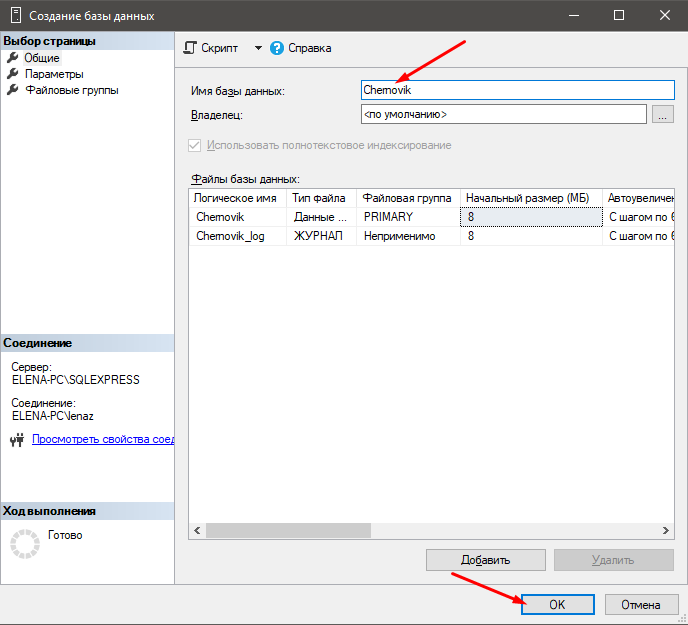


Рисунок 5 - Окно создания базы данных

В результате ваша созданная база данных «Chernovik» появится в папке «Базы данных» (рис.6)

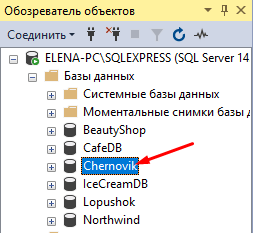


Рисунок 6 - Созданная база данных

После создания базы данных, необходимо зайти в папку с заданием и открыть файл со скриптом, который называется «ms» (рис.7), с помощью двойного нажатия левой кнопки мыши.

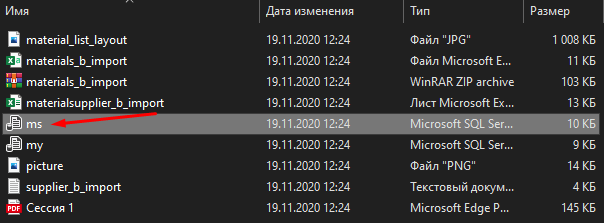


Рисунок 7 - Скрипт базы данных

После открытия файла появится код скрипта. Необходимо поменять базу данных, с которой будет работать скрипт, по умолчанию стоит база данных «master». Для этого необходимо в левом верхнем углу приложения найти название «master» и справа нажать стрелочку вниз (рис.8)

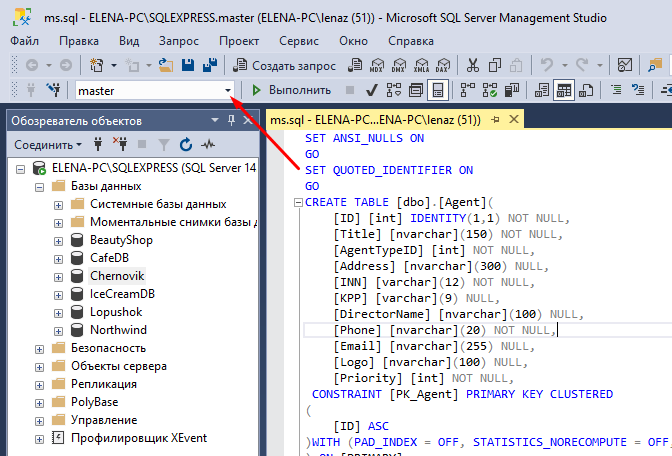


Рисунок 8 - Выбор базы данных

В появившемся списке нужно выбрать ранее созданную базу данных «Chernovik» (рис.9), нажав левую кнопку мыши.

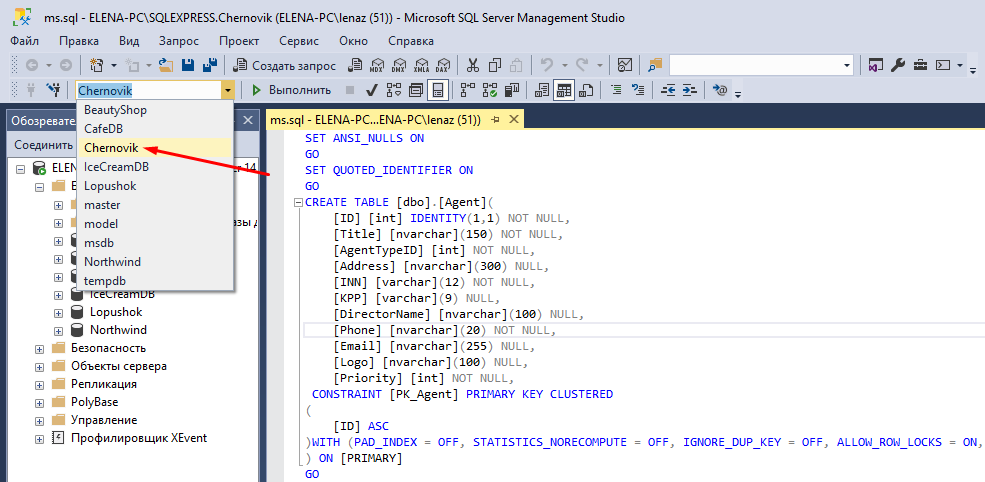


Рисунок 9 - Выбор базы данных

После выбора базы данных, нужно выполнить скрипт, нажав кнопку «Выполнить» (рис.10)

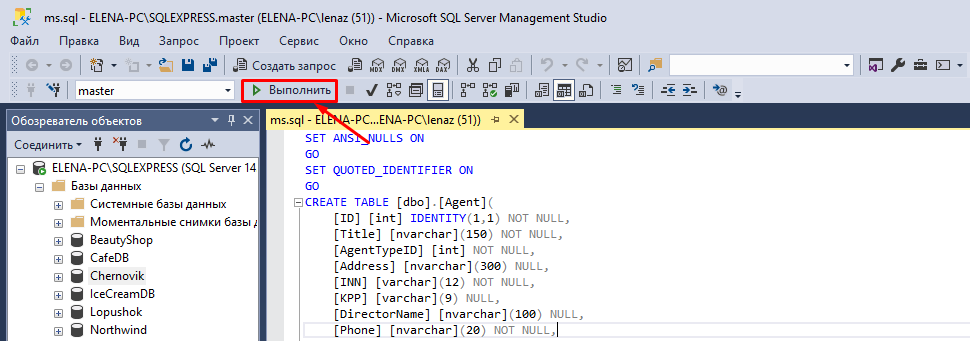


Рисунок 10 - Выполнение скрипта

В результате выполнения данного скрипта в ранее созданной базе данных появятся таблицы (рис.11): Agent, AgentPriorityHistory, AgentType, Material, MaterialCountHistory, MaterialSupplier, MaterialType, Product, ProductCostHistory, ProductMaterial, ProductSale, ProductType, Shop, Supplier. Некоторые из данных необходимо будет заполнить с помощью файлов в папке задания с пометкой «b\_import». Таких файлов в папке 3 (рис.12): materials\_b\_import, materialsupplier\_b\_import, supplier\_b\_import.

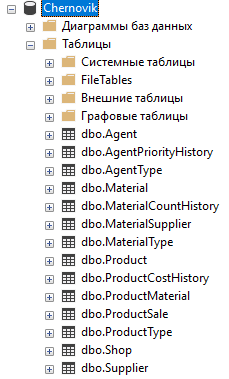


Рисунок 11 - Созданные таблицы

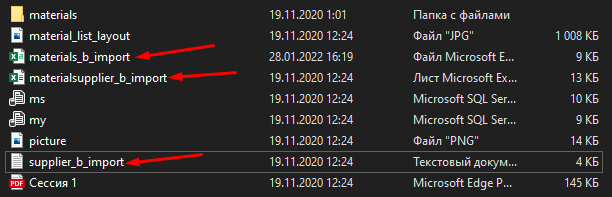


Рисунок 12 - Файлы для импорта

Начнём с файла «materials\_b\_import». Открываем файл двойным нажатием левой кнопки мыши. Откроется таблица с данными (рис.13), которые нужно обработать, чтобы далее вставить их в таблицу.

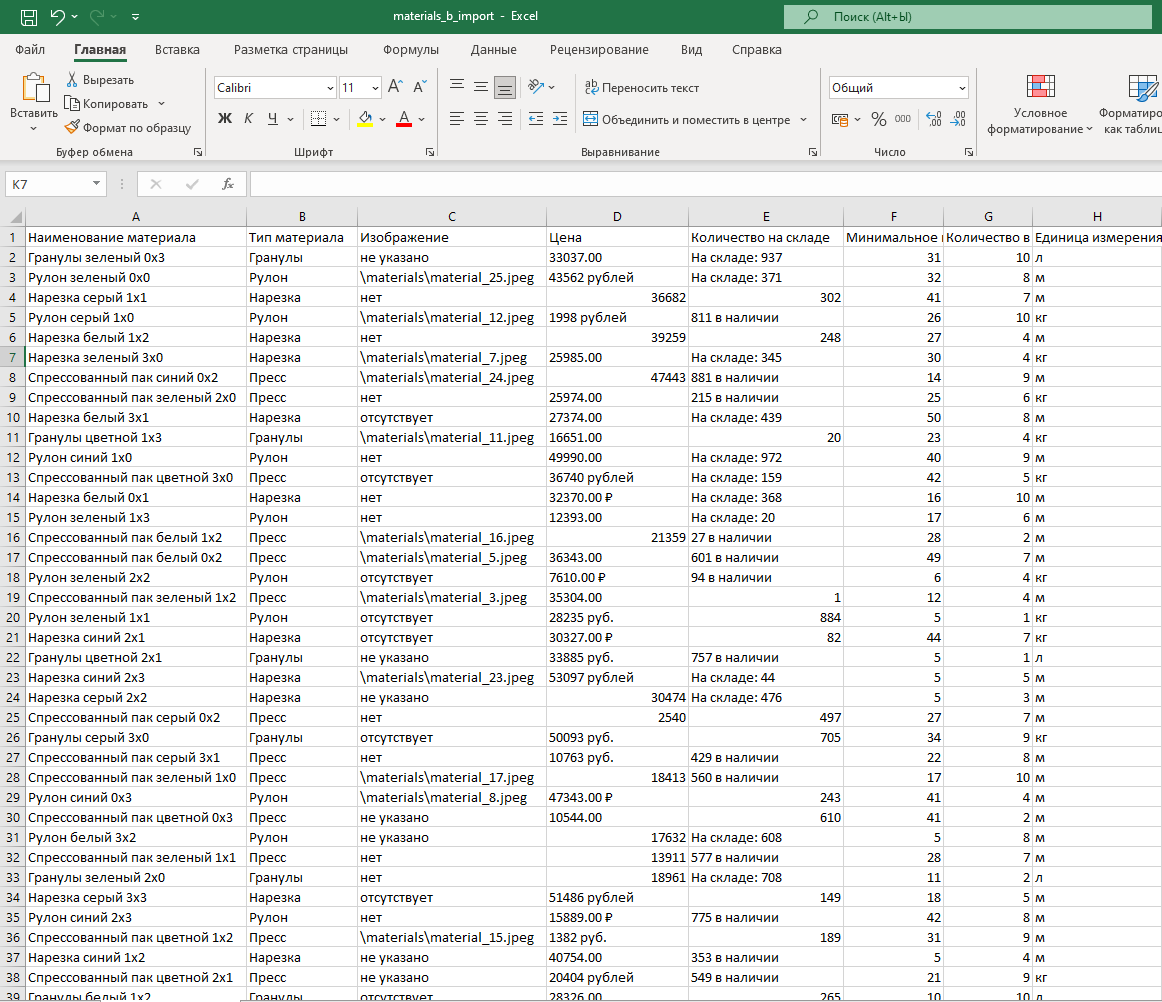


Рисунок 13 - Данные файла materials\_b\_import

Выделяем столбец «Изображение», в правом верхнем углу ищем изображение лупы, нажимаем его и выбираем кнопку «Заменить…» (рис.14)

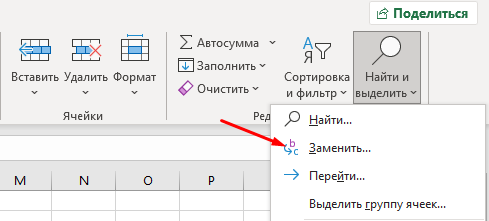


Рисунок 14 - Кнопка "Заменить..."

В появившемся окне «Найти и заменить» в поле «Найти» нужно написать «не указано» и нажать кнопку «Заменить все» (рис.15)

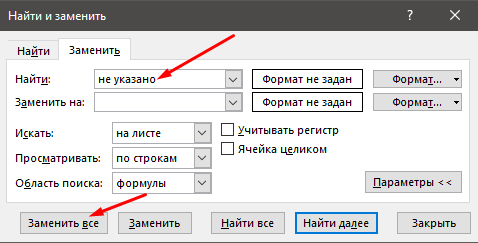


Рисунок 15 - Окно "Найти и заменить"

После нажатия кнопки «Заменить все» высветится окно с информацией, что записи были заменены (рис.16). Для продолжения работы нажимает на кнопку «ОК»

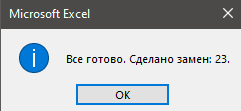


Рисунок 16 - Окно информации

Повторяем действия для этого же столбца, но в поле «Найти» пишем «отсутствует» (рис.17) и «нет» (рис.18)

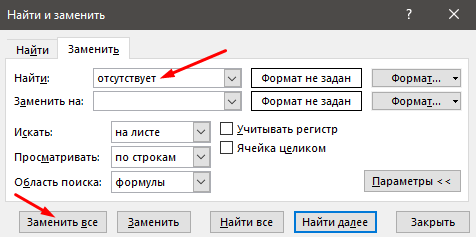


Рисунок 17 - Найти "отсутствует"

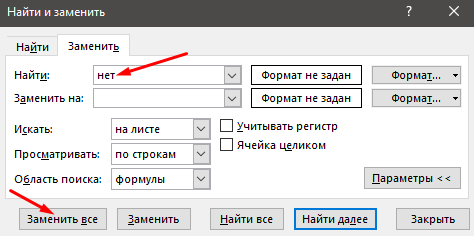


Рисунок 18 - Найти "нет"

Далее выделяем столбец «Цена» и опять нажимаем «Заменить…». В поле «Найти» пишем « рублей» (рис.19), «.00» (рис.20) и « руб.» (рис.21)

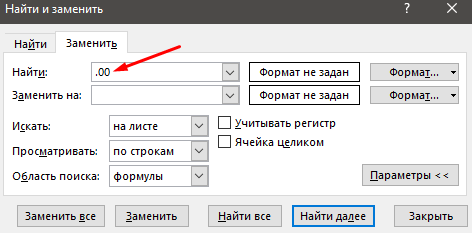


Рисунок 19 - Найти ".00"

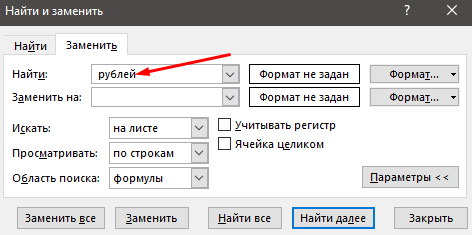


Рисунок 20 - Найти " рублей"

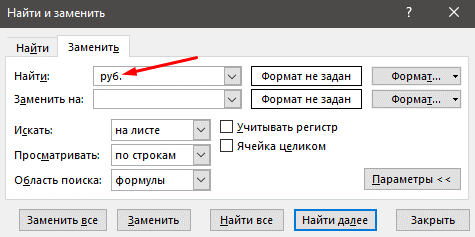


Рисунок 21 - Найти " руб."

Далее выделяем все значения столбца «Цена» и нажимаем правую кнопку мыши, в появившемся списке выбираем «Формат ячеек…»

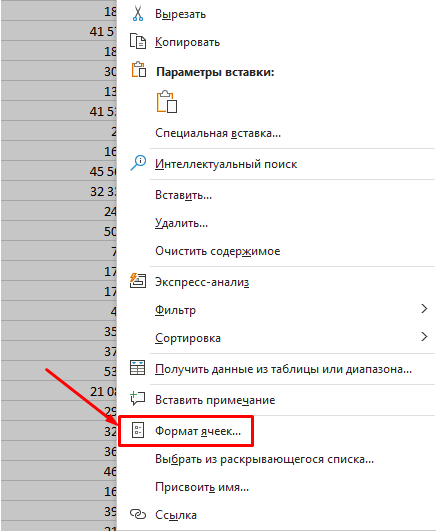


Рисунок 22 - Формат ячеек

В открывшемся окне «Формат ячеек» в списке «Числовые форматы» выбираем «Общий» и нажимаем кнопку «ОК» (рис.23)

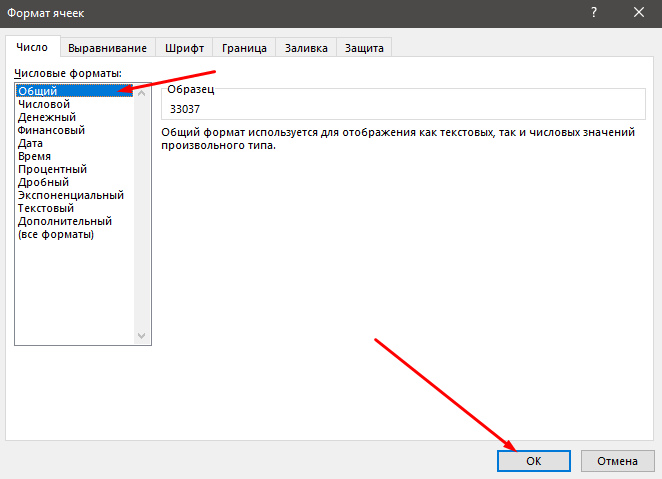


Рисунок 23 - Окно "Формат ячеек"

Выделяем столбец «Количество на складе» и нажимаем кнопку «Заменить…». В поле «Найти» пишем «На складе: » (рис.24) и « в наличии» (рис.25)

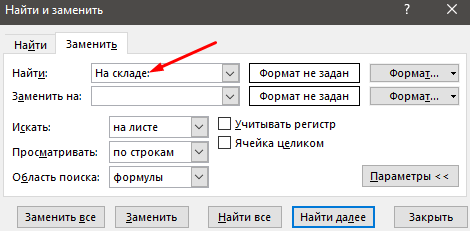


Рисунок 24 - Найти "На складе: "

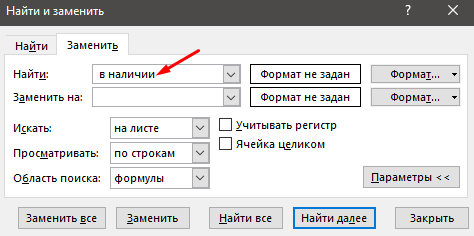


Рисунок 25 - Найти "в наличии"

Далее переходим на вкладку «Данные» (рис.26) и выбираем «Из таблицы/диапазона» (рис.27)

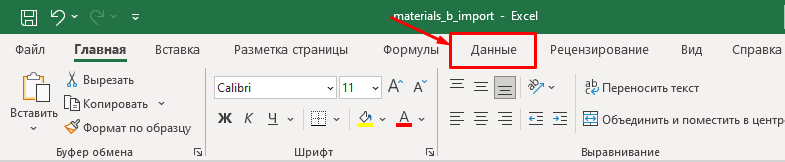


Рисунок 26 - Вкладка "Данные"

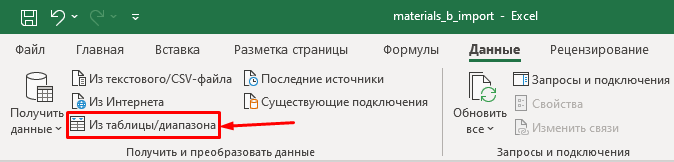


Рисунок 27 - "Из таблицы/диапазона"

Выделяем нашу таблицу (рис.28), ставим галочку для «Таблица с заголовками» и нажимаем кнопку «ОК».

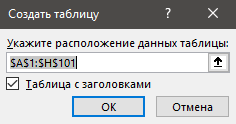


Рисунок 28 - Расположение таблицы

После нажатия кнопки «ОК» откроется «Редактор Power Query», в котором отобразится наша таблица. В таблице, находящейся в базе данных, присутствует столбец «MaterialTypeID», а в таблице для импорта – «Тип материала». Для получения ID необходимо создать таблицу с типами материалов. Для этого удаляем все столбцы, кроме «Тип материала». Чтобы это сделать нужно правой кнопкой нажать на столбец и выбрать «Удалить другие столбцы» (рис.29)

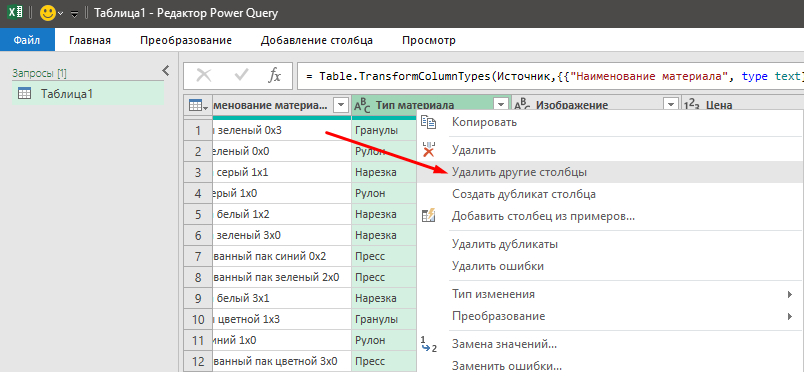


Рисунок 29 - Удаление других столбцов

После того, как остался один столбец, нужно правой кнопкой нажать по названию столбца и выбрать «Удалить дубликаты» (рис.30)

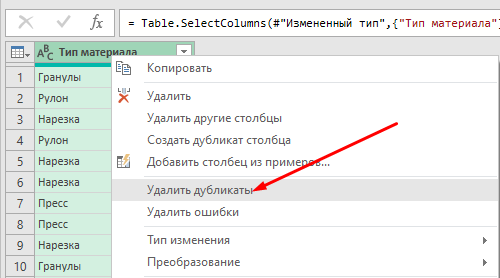


Рисунок 30 - Удаление дубликатов в столбце

После удаления дубликатов нам необходимо создать столбец с индексами. Для этого открываем вкладку «Добавление столбца» (рис.31)

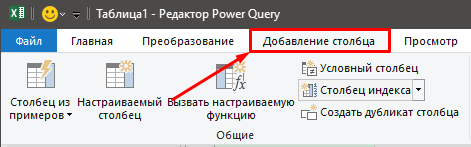


Рисунок 31 - Добавление столбца индекса

Ищем надпись «Столбец индекса» и нажимаем на стрелочку рядом с надписью. Развернётся список, в котором нужно выбрать «От 1» (рис.32)

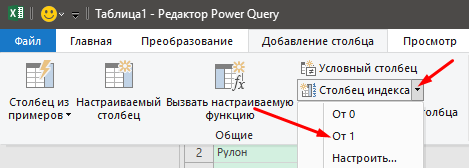


Рисунок 32 - Столбец индекса от 1

Появившейся столбец перетаскиваем налево, зажав на названии столбца левую кнопку мыши (рис.33)

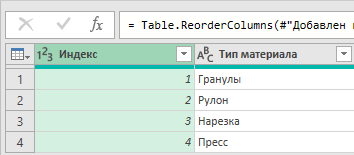


Рисунок 33 - Столбец индекса

Переходим на вкладку «Главная» и нажимаем на кнопку «Закрыть и загрузить» (рис.34)

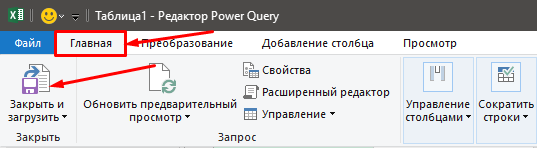


Рисунок 34 - Сохранение таблицы

После сохранения отобразится таблица (рис.35), в которой нужно добавить столбец «Процент брака». Для этого в третьем столбце со второй строки пишем нули (рис.36). Название столбца можно не изменять.



Рисунок 35 - Созданная таблица



Рисунок 36 - Столбец с нулями

Теперь копируем значения таблицы со второй строки (рис.37) (выделяем значения и либо нажимаем комбинацию «ctrl + c», либо правую кнопки мыши и «Копировать»)

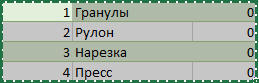


Рисунок 37 - Выделенные строки

Нужно вставить скопированное значение в таблицу «MaterialType». Для этого правой кнопкой мыши нажимаем по таблице «MaterialType» и выбираем «Изменить первые 200 строк» (рис.38)

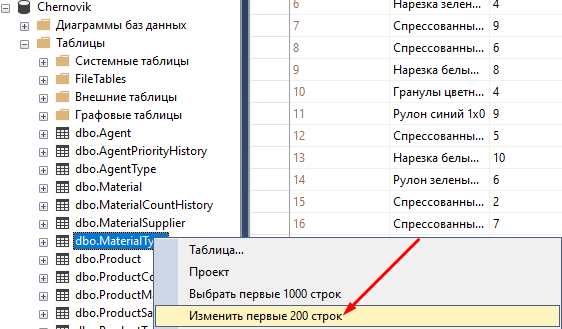


Рисунок 38 - Изменение таблицы "MaterialType"

Выделяем все столбцы при помощи стрелочки со звёздочкой (рис.39) и вставить ранее скопированные данные из таблицы (с помощью нажатия комбинации «ctrl + v» или нажав правую кнопку мыши и «Вставить»)



Рисунок 39 - Выделение строк

Возвращаемся к первоначальной таблице «materials\_b\_import», снова переходим в «PowerQuery».

Теперь нужно расставить столбцы по порядку, в котором они расположены в таблице «Material», находящейся в базе данных «Chernovik». Для этого в списке таблиц базы данных «Chernovik» необходимо правой кнопкой нажать по таблице «Material» и в появившемся списке выбираем «Изменить первые 200 строк» (рис.40)

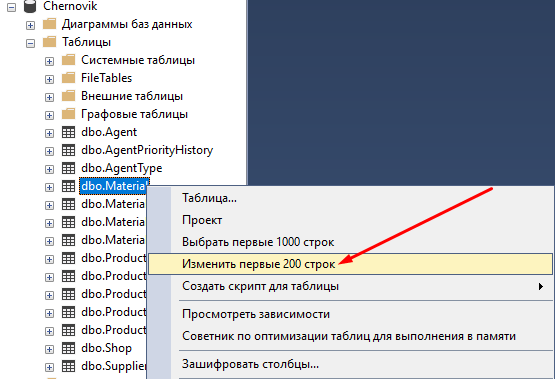


Рисунок 40 - Изменить первые 200 строк

Столбцы в PowerQuery нужно расставить как на рисунке 41 (Наименование материала, Количество в упаковке, Единица измерения, Количество, Минимальное количество, Цена, Описание, Изображение).



Рисунок 41 - Расположение столбцов

Чтобы создать столбец «Описание» нужно правой кнопкой нажать по столбцу «Изображение» и выбрать «Разделить столбец» => «По разделителю…» (рис.)

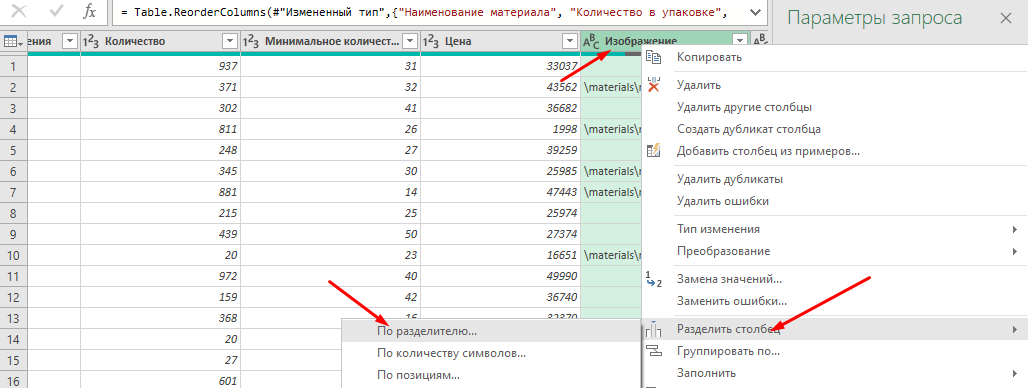


Рисунок 42 - Разделение столбца

В появившемся окне в поле «Выберите или введите разделитель» выбираем «--Пользовательский--» (рис.43)

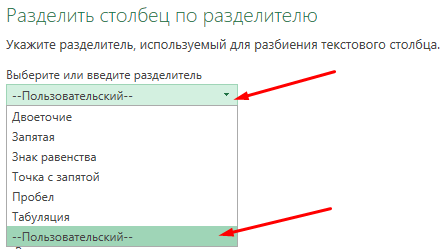


Рисунок 43 - Разделитель "Пользовательский"

В поле для разделителя пишем «/», выбираем разделение «Самый левый разделитель», символ кавычек – «Нет», а после нажимаем кнопку «ОК» (рис.44)

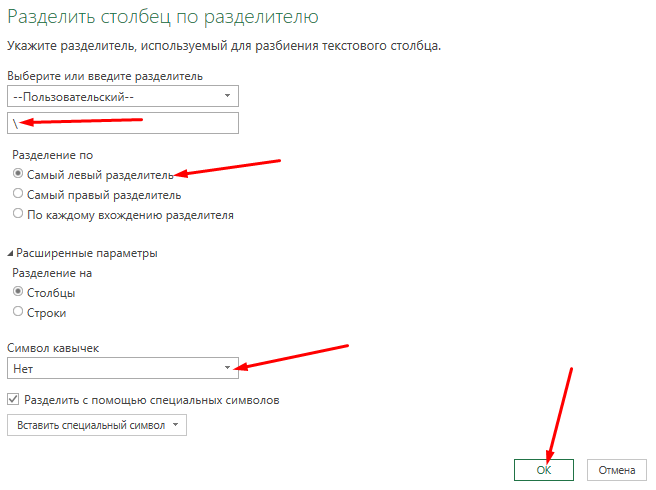


Рисунок 44 – Окно разделения столбца

После перетаскиваем столбец налево от столбца «Изображение». Далее нужно создать столбец с индексом, как делали до этого, и перетащить его в самое лево. Теперь необходимо сделать столбец «MaterialTypeID». Для этого на вкладке «Главная» нажать «Объединить запросы» (рис.45)

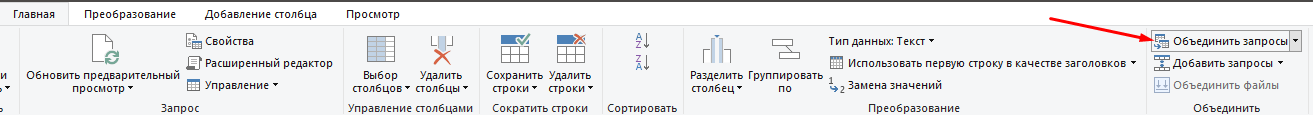


Рисунок 45 - Объединить запросы

В появившемся окне в «Таблица 1» необходимо выбрать столбец «Тип материала», в поле под таблицей выбрать таблицу с типами материалов, которую создали ранее, и в данной таблице выделить столбец «Тип материала». После всех выделений нажать кнопку «ОК» (рис.46)

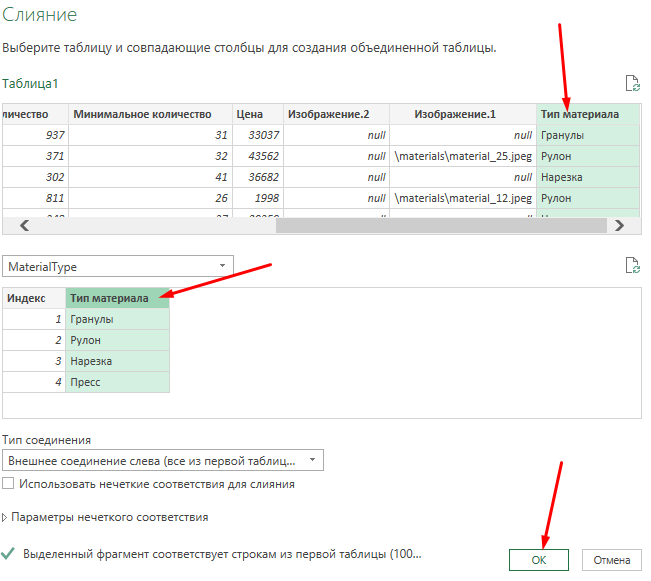


Рисунок 46 - Окно слияния

После слияния в появившемся столбце нужно нажать на «стрелочки» и там оставить галочку у «Индекс» и нажать кнопку «ОК» (рис.47)

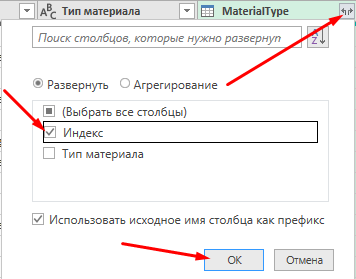


Рисунок 47 - Настройка столбца

Теперь надо удалить столбец «Тип материала». Для этого правой кнопкой нужно нажать по столбцу и выбрать «Удалить» (рис.48)

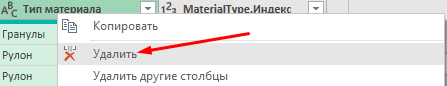


Рисунок 48 - Удаление столбца

Сохраняем таблицу с помощью кнопки «Закрыть и загрузить» на вкладке «Главная».

Копируем записи таблицы (либо нажимаем комбинацию «ctrl + c», либо правую кнопки мыши и «Копировать»), не включая названия столбцов, и, нажав правую кнопку мыши по таблицу «Material» в базе данных «Chernovik», выбрать «Изменить первые 200 строк». В появившемся окне выбрать все записи при помощи стрелочки со звёздочкой и вставить скопированные данные с помощью комбинации «ctrl + v» или нажатия правой кнопкой мыши => «Вставить».

Теперь нужно импортировать данные из файла «supplier\_b\_import». Для этого нужно открыть приложение «Excel» (рис.49) и создать пустую книгу.



Рисунок 49 - Приложение Excel

Теперь нужно перейти на вкладку «Данные» и нажать «Из текстового/CSV-файла» (рис.50)

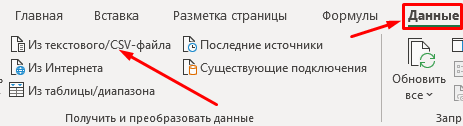


Рисунок 50 - Получение данных из текстового файла

В открывшемся окне нужно открыть папку с заданием и в ней выбрать файл «supplier\_b\_import», а после нажать кнопку «ИМПОРТ»

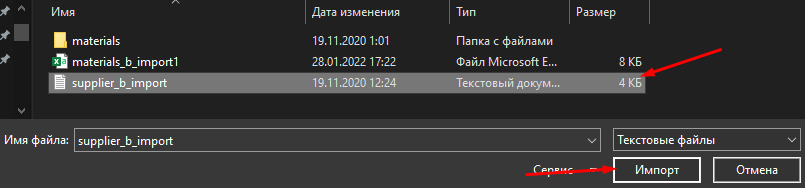


Рисунок 51 - Выбор файла

После нажатия в поле «Источник файла» нужно выбрать «65001: Юникод (UTF-8)», а после нажать на кнопку «Преобразовать данные» (рис.52)

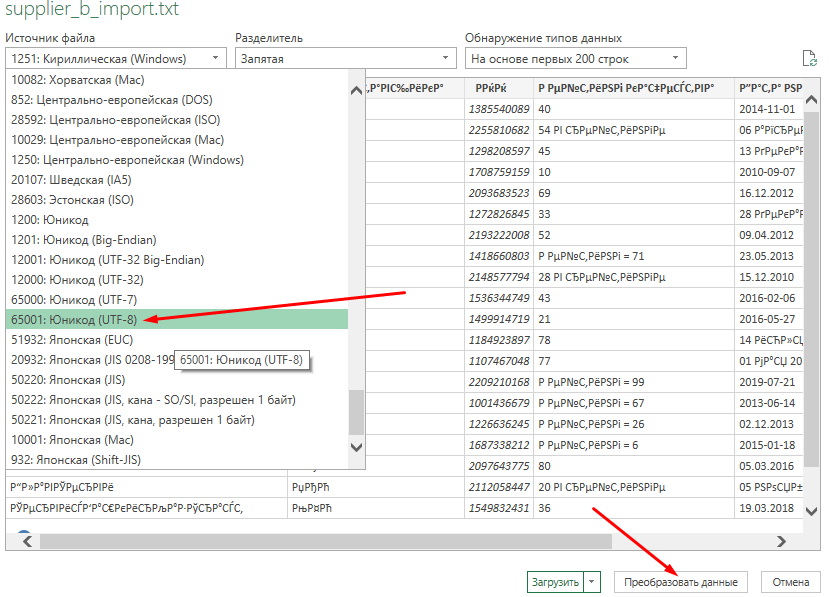


Рисунок 52 - Преобразование данных

Сохраняем таблицу с помощью кнопки «Закрыть и загрузить» на вкладке «Главная».

Выделяем столбец «Рейтинг качества» и в правом верхнем углу нажимаем на «Найти и выделить» => «Заменить…». В появившемся окне в поле «Найти» пишем «Рейтинг = » (рис.53) и « в рейтинге» (рис.54) и нажимаем «Заменить все».

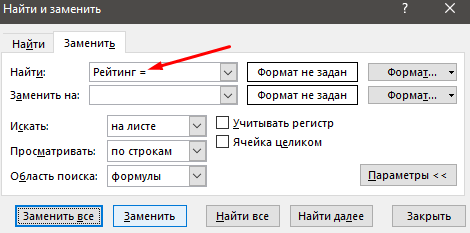


Рисунок 53 - Найти "Рейтинг ="

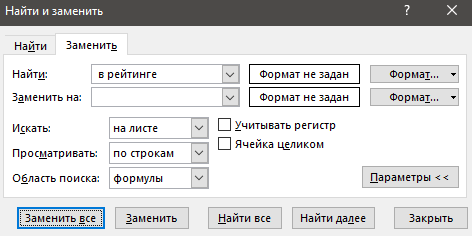


Рисунок 54 - Найти "в рейтинге"

Далее переходим на вкладку «Данные» (рис.26) и выбираем «Из таблицы/диапазона». Переставляем столбцы в порядке, как в базе данных (Наименование поставщика, ИНН, Дата начала работы с поставщиком, Рейтинг качества, Тип поставщика). Добавляем столбец индекса с помощью вкладки «Добавление столбца» => «Столбец индекса» => «От 1», переносим столбец в самое лево.

Сохраняем таблицу с помощью кнопки «Закрыть и загрузить» на вкладке «Главная».

Выделяем значения таблицы, копируем их с помощью комбинации «ctrl + c» или нажатия правой кнопкой мыши => «Копировать». В базе данных правой кнопкой нажимаем по таблице «Supplier» => «Изменить первые 200 строк». Выделяем все столбцы, нажав стрелочку со звёздочкой, и вставляем скопированные ранее данные, нажав комбинацию «ctrl + v» или правую кнопку мыши => «Вставить»

Далее переходим к файлу «materialsupplier\_b\_import». Открываем файл, переходим к вкладке «Данные» => «Получить данные» => «Из книги» (рис.55)

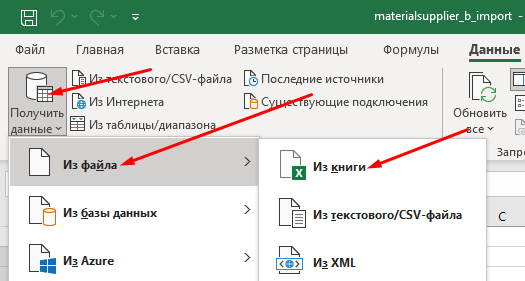


Рисунок 55 - Открытие других книг

Открываем ранее отредактированные нами файлы «supplier\_b\_import» и «materials\_b\_import».

Выбираем таблицу файла «materialsupplier\_b\_import», на вкладке «Главная» нажимаем «Объединение запросов». Выделяем столбец «Наименование материала» в таблице файла «materialsupplier\_b\_import», далее выделяем столбец «Наименование материала» в таблице файла «materials\_b\_import» (рис.56) и нажимаем кнопку «ОК»

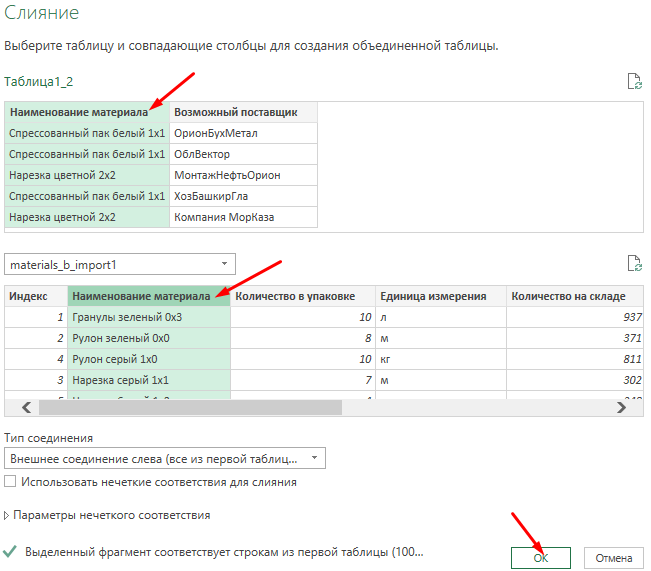


Рисунок 56 – Слияние с материалами

После слияния в появившемся столбце нужно нажать на «стрелочки» и оставить галочку только у «Индекс» и нажать кнопку «ОК»

На вкладке «Главная» опять нажимаем «Объединение запросов». Выделяем столбец «Возможный поставщик» в таблице файла «materialsupplier\_b\_import», далее выделяем столбец «Наименование поставщика» в таблице файла «materials\_b\_import» (рис.57) и нажимаем кнопку «ОК»

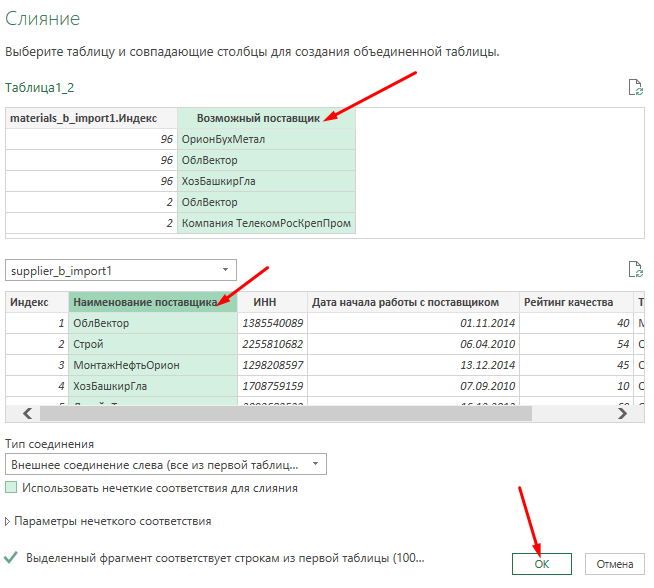


Рисунок 57 - Слияние с поставщиками

После слияния в появившемся столбце нужно нажать на «стрелочки» и оставить галочку у «Индекс» и нажать кнопку «ОК»

Сохраняем таблицу с помощью кнопки «Закрыть и загрузить» на вкладке «Главная».

Выделяем значения таблицы, копируем их с помощью комбинации «ctrl + c» или нажатия правой кнопкой мыши => «Копировать». В базе данных правой кнопкой нажимаем по таблице «MaterialSupplier» => «Изменить первые 200 строк». Выделяем все столбцы, нажав стрелочку со звёздочкой, и вставляем скопированные ранее данные, нажав комбинацию «ctrl + v» или правую кнопку мыши => «Вставить».

Запускаем приложение «Visual Studio» (рис.58)



Рисунок 58 - Приложение Visual Studio

В открывшемся приложении нажимаем на кнопку «Создание проекта». Далее выбираем шаблон проекта «Приложение WPF (.NET Framework)» (рис.59) и нажимаем кнопку далее.



Рисунок 59 - Шаблон проекта

В следующем окне вводим название проекта (Chernovik) (рис.60) (также в этом окне можно поменять расположение файла, имя решения и платформу) и жмём кнопку «Создать».

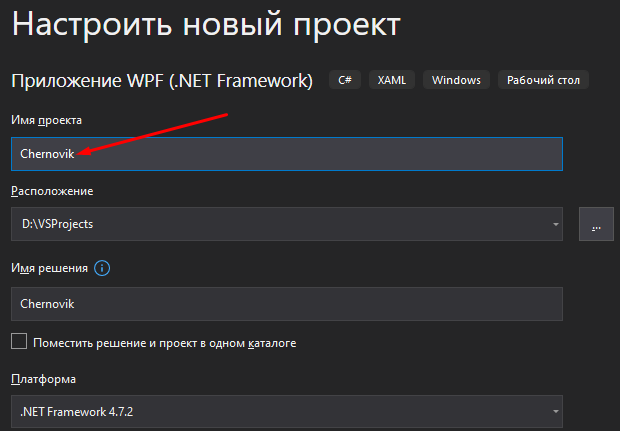


Рисунок 60 - Настройка проекта

После создания можно увидеть поле, в которое нужно будет вводить код, влияющий на внешний вид формы, которая находится над самим полем (рис.61)

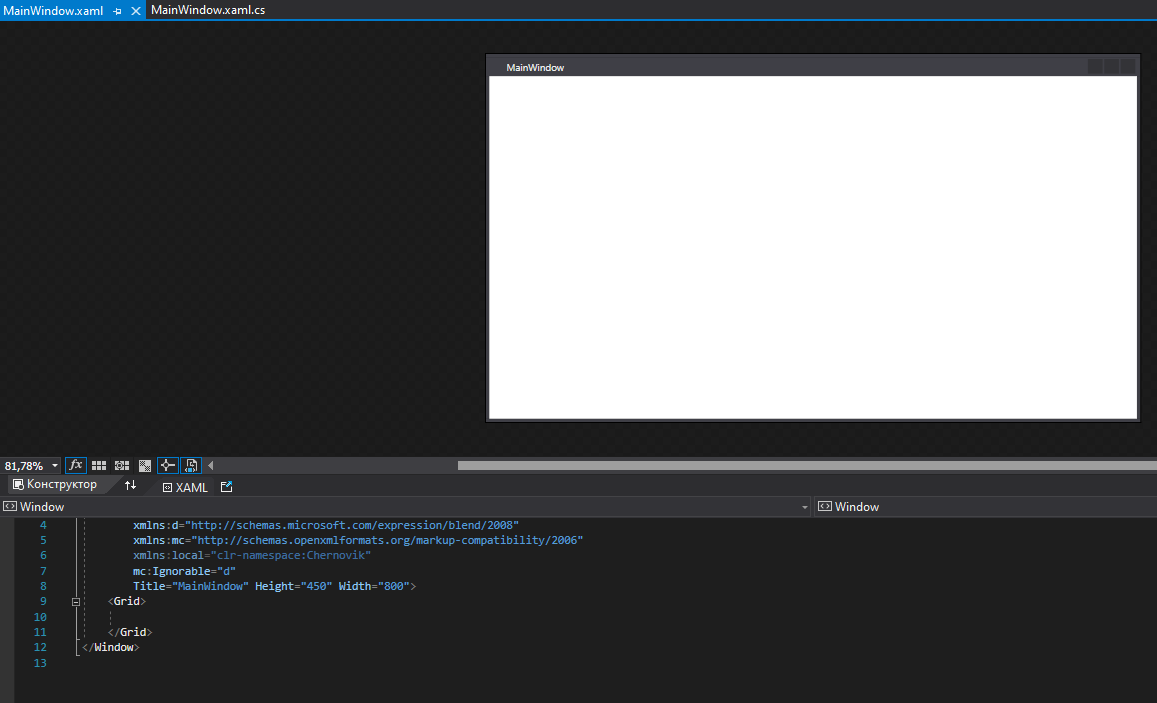


Рисунок 61 - Внешний вид программы

Находим строчку кода Title="MainWindow" Height="450" Width="800"> и изменяем на Title="Черновик" Height="700" Width="900">. В данной строчке мы поменяли название формы и её высоту и ширину.

С помощью строк кода создаём «полоску» сверху окна, чтобы потом поместить туда логотип и кнопку «Назад». В коде настраивается размер данной «полоски»

<Grid.RowDefinitions>

<RowDefinition Height="20\*"/>

<RowDefinition Height="200\*"/>

</Grid.RowDefinitions>

Создаём папку «Resources», в которую поместим логотип.

Далее вставляем логотип и кнопку «Назад». В коде кнопки нужно создать обработчик событий «Click».

<StackPanel Grid.Row="0"

Background="#FEA9AB"

Orientation="Horizontal">

<Image Source=".//Resources/Черновик.png"

Width="55"

HorizontalAlignment="Left"

Margin="10 0 0 0"/>

<Button x:Name="BackBtn"

Click="BackBtn\_Click"

Width="150" Height="30"

FontFamily="Segoe Print" FontSize="17"

Foreground="White" Background="#EAB14D"

Margin="650 0 0 0">Назад</Button>

</StackPanel>

Создаём папку «DataBase», в которую добавляем модель базы данных. Для этого правой кнопкой нажимаем по созданной папке, выбираем «Добавить» => «Добавить элемент…» (рис.62)

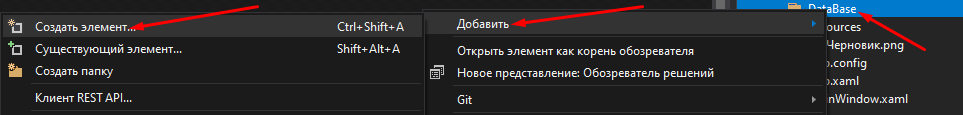


Рисунок 62 - Добавление модели

В появившемся окне слева выбираем «Данные», по середине в списке находим «Модель ADO.NET EDM» и нажимаем кнопку «Добавить» (рис.63)

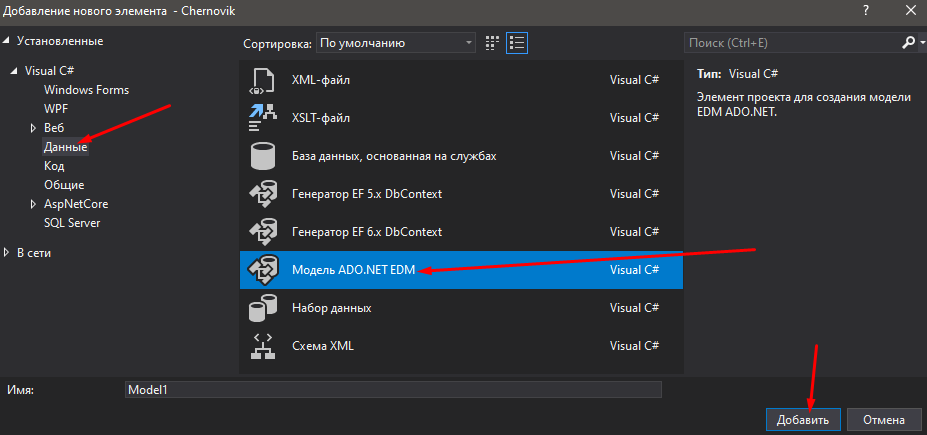


Рисунок 63 - Добавление Модели ADO.NET EDM

В следующем окне просто нажимаем кнопку «Далее». В окне подключения нажимаем кнопку «Создать соединение…» (рис.64)

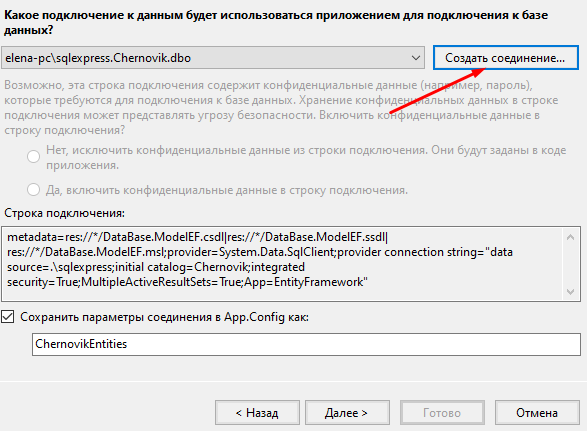


Рисунок 64 - Создание соединения

В поле «Имя сервера» вводим «.\sqlexpress», снизу выбираем базу данных «Chernovik» и нажимаем кнопку «ОК» (рис.65)

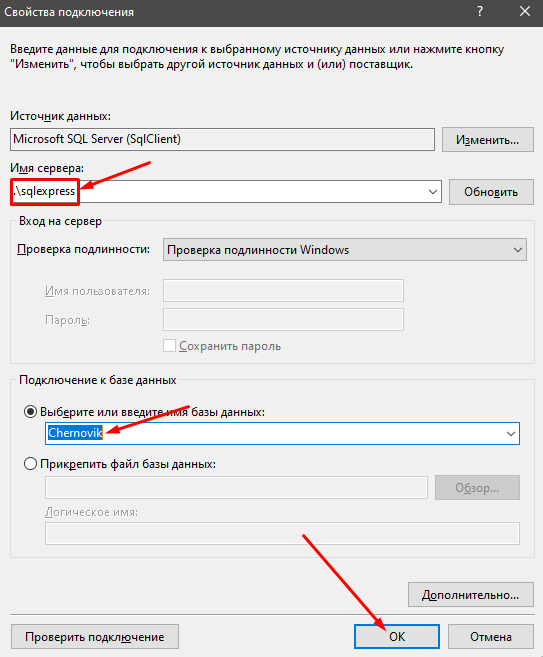


Рисунок 65 – Свойства подключения

Нажимаем кнопку «Далее» 2 раза. В окне выбора параметров и объектов базы данных ставим галочку напротив «Таблицы» и ждём кнопку «Готово» (рис.66)

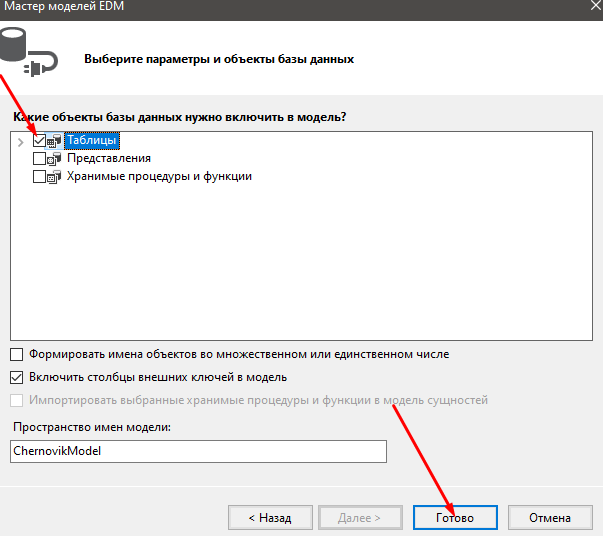


Рисунок 66 - Выбор таблиц

В папке «DataBase» создаём класс «Transition» (рис.67)

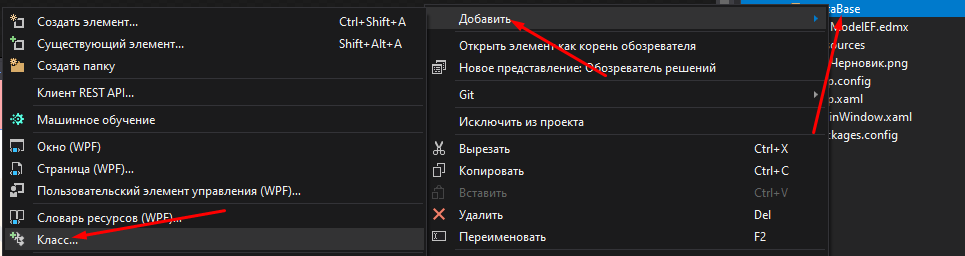


Рисунок 67 - Создание класса

В классе прописываем код для создания объекта для перехода по формам

public static Frame MainFrame { get; set; }

Также создаём объект для передачи данных из базы данных

private static ChernovikEntities context;

public static ChernovikEntities Context

{

get

{

if (context == null)

context = new ChernovikEntities();

return context;

}

}

Создаём папку «Pages», в которую добавляем страницу с названием «MaterialList».

Переходим к коду окна «MainWindow» и пишем строки кода, чтобы в данном окне показывалась страница «MaterialList»

FrmMain.Navigate(new MaterialList());

Transition.MainFrame = FrmMain;

Для события «BackBtn\_Click» пишем одну строчку кода, чтобы кнопка возвращала на предыдущую страницу:

Transition.MainFrame.GoBack();

В дизайнере страницы «MaterialList» меняем название и размеры страницы:

d:DesignHeight="700" d:DesignWidth="900"

Title="Список материалов"

И добавляем событие изменения видимости:

IsVisibleChanged="Page\_IsVisibleChanged"

В файле «App.xaml» назначаем стили для основных элементов.

Для кнопки:

<Style TargetType="Button" x:Key="mainBtn">

<Setter Property="Width" Value="200"/>

<Setter Property="Height" Value="35"/>

<Setter Property="Margin" Value="5"/>

<Setter Property="FontFamily" Value="Segoe Print"/>

<Setter Property="FontSize" Value="17"/>

<Setter Property="Foreground" Value="White"/>

<Setter Property="Background" Value="#EAB14D"/>

</Style>

Для надписи:

<Style TargetType="TextBlock" x:Key="mainTxtb">

<Setter Property="HorizontalAlignment" Value="Center"/>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontSize" Value="17"/>

<Setter Property="FontFamily" Value="Segoe Print"/>

<Setter Property="TextAlignment" Value="Right"/>

<Setter Property="Width" Value="200"/>

</Style>

Для строки ввода текста:

<Style TargetType="TextBox" x:Key="mainTxt">

<Setter Property="HorizontalAlignment" Value="Center"/>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Segoe Print"/>

<Setter Property="FontSize" Value="15"/>

<Setter Property="Height" Value="30"/>

<Setter Property="Width" Value="250"/>

<Setter Property="Margin" Value="10"/>

<Setter Property="Padding" Value="0,4"/>

</Style>

Для выпадающего списка:

<Style TargetType="ComboBox" x:Key="mainCombo">

<Setter Property="HorizontalAlignment" Value="Center"/>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Segoe Print"/>

<Setter Property="FontSize" Value="15"/>

<Setter Property="Foreground" Value="Black"/>

<Setter Property="Width" Value="200"/>

<Setter Property="Height" Value="30"/>

<Setter Property="Margin" Value="10"/>

</Style>

Для check box:

<Style TargetType="CheckBox" x:Key="mainCheck">

<Setter Property="FontFamily" Value="Segoe Print"/>

<Setter Property="FontSize" Value="15"/>

</Style>

Для изображения:

<Style TargetType="Image" x:Key="ImgSource">

<Setter Property="Width" Value="150"/>

<Setter Property="HorizontalAlignment" Value="Center"/>

</Style>

Возвращаемся в дизайнер страницы «MaterialList». Добавляем код для «полоски» сверху, на которой будет находится сортировка, фильтрация и поиск по названию и описанию материала.

Назначаем размер «полоски»:

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition Width="325"/>

<ColumnDefinition Width="240"/>

</Grid.ColumnDefinitions>

Добавляем строку поиска:

<TextBox HorizontalAlignment="Left"

VerticalAlignment="Bottom"

Style="{StaticResource mainTxt}"

Text="Введите для поиска"

x:Name="SearchMaterialBox" TextChanged="SearchMaterialBox\_TextChanged"

LostFocus="SearchMaterialBox\_LostFocus" GotFocus="SearchMaterialBox\_GotFocus"

Margin="10,0,0,10" Width="305"/>

Добавляем сортировку материалов по наименованию, остатку на складе и стоимости:

<ComboBox x:Name="SortCBox" SelectionChanged="SortCBox\_SelectionChanged"

Style="{StaticResource mainCombo}" Width="180">

<ComboBoxItem>

<TextBlock Text="Нет"

Style="{StaticResource mainTxtb}"

Width="160"/>

</ComboBoxItem>

<ComboBoxItem>

<TextBlock Text="Наименование"

Style="{StaticResource mainTxtb}"

Width="150"/>

</ComboBoxItem>

<ComboBoxItem>

<TextBlock Text="Остаток на складе"

Style="{StaticResource mainTxtb}"

Width="150"/>

</ComboBoxItem>

<ComboBoxItem>

<TextBlock Text="Стоимость"

Style="{StaticResource mainTxtb}"

Width="150"/>

</ComboBoxItem>

Добавляем check box, который будет отвечать за сортировку по убыванию:

<CheckBox Content="Убывание" VerticalAlignment="Center"

x:Name="CheckSort"

Checked="CheckSort\_Checked" Unchecked="CheckSort\_Unchecked"

Style="{StaticResource mainCheck}" Width="100"/>

Добавляем фильтрацию по типу материала:

ComboBox Grid.Column="2" DisplayMemberPath="Title"

x:Name="MaterialTypeBox" SelectionChanged="MaterialTypeBox\_SelectionChanged"

Style="{StaticResource mainCombo}" Margin="90,10,10,10"/>

</Grid>

Добавляем ListView, который будет отображать записи из базы данных.

<Grid Grid.Row="1">

<Grid.RowDefinitions>

<RowDefinition Height="25\*"/>

</Grid.RowDefinitions>

<ScrollViewer>

<WrapPanel Orientation="Vertical">

<ListView HorizontalAlignment="Center"

VerticalAlignment="Center"

Margin="90 0 0 80"

x:Name="ListMaterial"

Background="Transparent" Width="700"

BorderBrush="Transparent"

SelectionChanged="ListMaterial\_SelectionChanged">

Задаём стиль каждого контейнера ListView.

<ListView.ItemContainerStyle>

<Style TargetType="ListViewItem">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="ListBoxItem">

<Border BorderBrush="Black"

Background="Transparent"

Width="670"

Height="150"

x:Name="BorderTest"

BorderThickness="1"

Margin="0 10 0 0">

Каждый контейнер будет содержать изображение:

<Border BorderBrush="Black"

Background="Transparent">

<WrapPanel Orientation="Vertical"

Width="150">

<Image Margin="10 10 10 10"

Grid.Row="0"

Source="{Binding Image, FallbackValue={StaticResource Null}}"

Width="130"/>

</WrapPanel>

</Border>

Тип материала:

<WrapPanel Orientation="Horizontal">

<TextBlock Text="{Binding MaterialType.Title}"

Margin="10 10 0 0"

x:Name="TxtMaterialType"

Style="{StaticResource materialTxtb}"

FontSize="19"/>

Название материала:

<TextBlock Text="{Binding Title}"

Margin="0 10 0 0"

x:Name="TxtNameMaterial"

Style="{StaticResource materialTxtb}"

FontSize="19"/>

Минимальное количество на складе:

<WrapPanel Orientation="Horizontal">

<TextBlock x:Name="Materials"

Text="Минимальное количество: "

Margin="10 0 0 0"

Style="{StaticResource materialTxtb}"/>

<TextBlock x:Name="MinCount"

Text="{Binding MinCount}"

Margin="5 0 0 0"

Style="{StaticResource materialTxtb}"/>

</WrapPanel>

Поставщиков:

<WrapPanel Orientation="Horizontal">

<TextBlock x:Name="TxtSupplier"

Text="Поставщики: "

Margin="10 0 0 0"

Style="{StaticResource materialTxtb}"/>

</WrapPanel>

Стоимость:

<WrapPanel Orientation="Horizontal">

<TextBlock x:Name="TxtCost"

Text="Стоимость: "

Margin="10 0 0 0"

Style="{StaticResource materialTxtb}"/>

<TextBlock x:Name="Cost"

Text="{Binding Cost}"

Margin="5 0 0 0"

Style="{StaticResource materialTxtb}"/>

</WrapPanel>

Чтобы при наведении и нажатии плитка меняла цвет необходимо прописать код:

<ControlTemplate.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter TargetName="BorderTest" Property="Background" Value="#FEA9AB"/>

<Setter TargetName="BorderTest" Property="Cursor" Value="Hand"/>

</Trigger>

<Trigger Property="IsSelected" Value="True">

<Setter TargetName="BorderTest" Property="Background" Value="#EAB14D"/>

</Trigger>

</ControlTemplate.Triggers>

В нижнюю часть приложения поместим 3 кнопки: добавление, изменение и удаление

<WrapPanel Grid.Row="2">

<Button Content="Добавить"

Style="{StaticResource mainBtn}"

x:Name="AddBtn" Click="AddBtn\_Click"/>

<Button Content="Изменить" Visibility="Hidden"

Style="{StaticResource mainBtn}"

x:Name="BtnEdit" Click="BtnEdit\_Click"/>

<Button Content="Удалить" Visibility="Hidden"

Style="{StaticResource mainBtn}"

x:Name="BtnDelete" Click="BtnDelete\_Click"/>

</WrapPanel>

Перейдём к коду страницы «MaterialList». Нам нужно создать переменную, которая будет хранить все значения таблицы «MaterialType»:

var filtItems = Transition.Context.MaterialType.ToList();

В выпадающий список передаём значения созданной переменной:

MaterialTypeBox.ItemsSource = filtItems;

Делаем так, чтобы в выпадающих списках показывалась самая первая запись:

MaterialTypeBox.SelectedIndex = 0;

SortCBox.SelectedIndex = 0;

Передадим в ListView данные из таблицы «Material»

ListMaterial.ItemsSource = Transition.Context.Material.ToList();

Создадим вспомогательный метод «DataView», к которому будут ссылаться обработчики событий

public void DataView()

В методе создадим переменную, содержащую данные из таблицы «Material»

var tempDataMaterial = Transition.Context.Material.ToList();

Напишем код, который будет искать в названии материала или в описании то, что написано в строке поиска, если она не пустая

if (SearchMaterialBox.Text != "Введите для поиска" && SearchMaterialBox.Text != null)

tempDataMaterial = tempDataMaterial

.Where(p => p.Title.ToLower().Contains(SearchMaterialBox.Text.ToLower())

|| p.Description.ToLower().Contains(SearchMaterialBox.Text.ToLower()))

.ToList();

Напишем код для сортировки, который будет сортировать и по убыванию, и по возрастанию.

switch (SortCBox.SelectedIndex)

{

case 1:

{

if (!(bool)CheckSort.IsChecked)

tempDataMaterial = tempDataMaterial

.OrderBy(p => p.Title)

.ToList();

else

tempDataMaterial = tempDataMaterial

.OrderByDescending(p => p.Title)

.ToList();

ListMaterial.ItemsSource = tempDataMaterial;

break;

}

case 2:

{

if (!(bool)CheckSort.IsChecked)

tempDataMaterial = tempDataMaterial

.OrderBy(p => p.CountInStock)

.ToList();

else

tempDataMaterial = tempDataMaterial

.OrderByDescending(p => p.CountInStock)

.ToList();

ListMaterial.ItemsSource = tempDataMaterial;

break;

}

case 3:

{

if (!(bool)CheckSort.IsChecked)

tempDataMaterial = tempDataMaterial

.OrderBy(p => p.Cost)

.ToList();

else

tempDataMaterial = tempDataMaterial

.OrderByDescending(p => p.Cost)

.ToList();

ListMaterial.ItemsSource = tempDataMaterial;

break;

}

}

Сделаем так, чтобы кнопки «Изменить» и «Удалить» появлялись только когда выбрана запись

if (ListMaterial.SelectedItems.Count > 0)

{

BtnDelete.Visibility = Visibility.Visible;

BtnEdit.Visibility = Visibility.Visible;

}

else

{

BtnDelete.Visibility = Visibility.Hidden;

BtnEdit.Visibility = Visibility.Hidden;

}

Создадим страницу с названием «AddMaterial», предназначенную для добавления нового материала.

Создадим выпадающий список с типом материала, поля для ввода наименования, описания, фотографии, минимального количества материала и количества материала на складе

<StackPanel HorizontalAlignment="Center"

VerticalAlignment="Center"

Orientation="Vertical" Height="auto" Width="auto">

<StackPanel Orientation="Horizontal">

<TextBlock Text="Тип материала: "

Style="{StaticResource mainTxtb}"/>

<ComboBox Name="MaterialTypeCombo"

DisplayMemberPath="Title" SelectedValuePath="ID" SelectionChanged="MaterialTypeCombo\_SelectionChanged"

Style="{StaticResource mainCombo}"

Width="250"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Наименование: "

Style="{StaticResource mainTxtb}"/>

<TextBox Name="TxtName"

Style="{StaticResource mainTxt}"/>

</StackPanel>

<StackPanel Orientation="Horizontal"/>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Описание: "

Style="{StaticResource mainTxtb}"/>

<TextBox Name="TxtDescription"

Style="{StaticResource mainTxt}"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Фотография: "

Style="{StaticResource mainTxtb}"/>

<TextBox Name="TxtImage"

Style="{StaticResource mainTxt}"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Минимальное количество: "

Style="{StaticResource mainTxtb}"/>

<TextBox Name="TxtMinCount"

Style="{StaticResource mainTxt}"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Количество на складе: "

Style="{StaticResource mainTxtb}"/>

<TextBox Name="TxtCountInStock"

Style="{StaticResource mainTxt}"/>

</StackPanel>

Добавим кнопку «Сохранить» для добавления нового материала в базу данных

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Center">

<Button Name="BtnSave"

Click="BtnSave\_Click"

Style="{StaticResource mainBtn}"

Content="Сохранить"/>

</StackPanel>

В коде страницы «MaterialList» для кнопки «AddBtn» добавим код перехода на страницу «AddMaterial»

Transition.MainFrame.Navigate(new AddMaterial());

В коде страницы «AddMaterial» добавим код, чтобы в выпадающий список попали значения типа материала из базы данных

var filtItems = Transition.Context.MaterialType.ToList();

filtItems.Insert(0, new MaterialType { Title = "Выберите тип" });

MaterialTypeCombo.ItemsSource = filtItems;

MaterialTypeCombo.SelectedIndex = 0;

Создадим вспомогательный метод, который будет менять ID типа материала на название этого типа

private void DataView()

{

var tempDataProduct = Transition.Context.Material.ToList();

if (MaterialTypeCombo.SelectedIndex > 0)

tempDataProduct = tempDataProduct.Where(p => p.MaterialTypeID == (MaterialTypeCombo.SelectedItem as MaterialType).ID).ToList();

}

Для кнопки «Сохранить» пишем код, который создаст новый экземпляр в таблице «Material» и сохранит значения из полей на странице в соответствующие поля таблицы в базе данных. Если не возникло никаких ошибок, выводится уведомление о сохранении, а если ошибки возникли выводится текст ошибки.

try

{

Material material = new Material()

{

Title = TxtName.Text,

MaterialType = MaterialTypeCombo.SelectedItem as MaterialType,

Description = TxtDescription.Text,

Image = TxtImage.Text,

MinCount = Convert.ToInt32(TxtMinCount.Text),

CountInStock = Convert.ToInt32(TxtCountInStock.Text),

};

Transition.Context.Material.Add(material);

Transition.Context.SaveChanges();

MessageBox.Show("Данные успешно добавлены", "Уведомление", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception er)

{

MessageBox.Show(er.Message.ToString());

}

Transition.MainFrame.GoBack();