

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

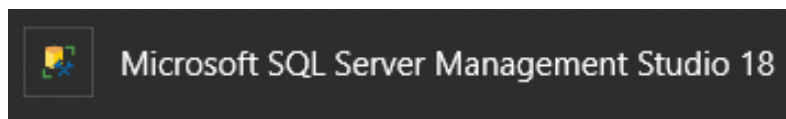


Рисунок 1 - Приложение SQL Server

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».

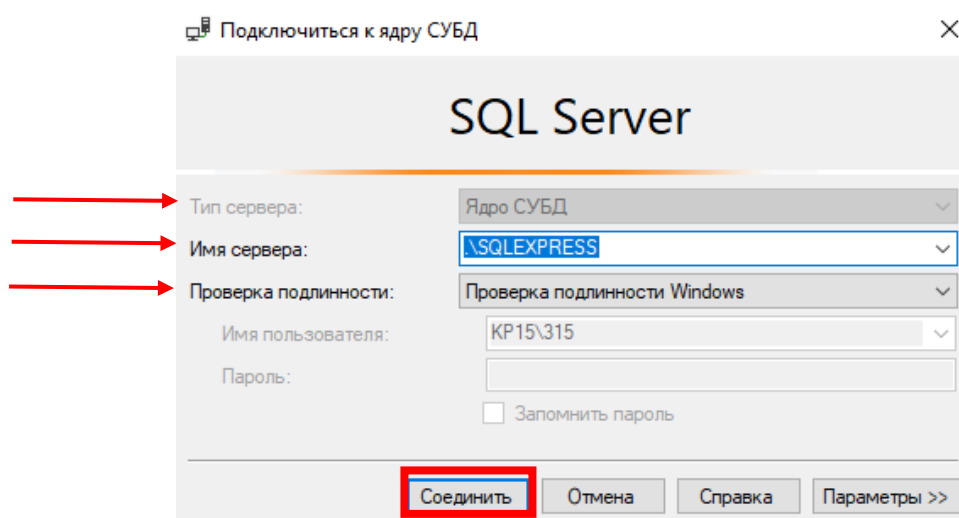


Рисунок 2 - Соединение с сервером

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.

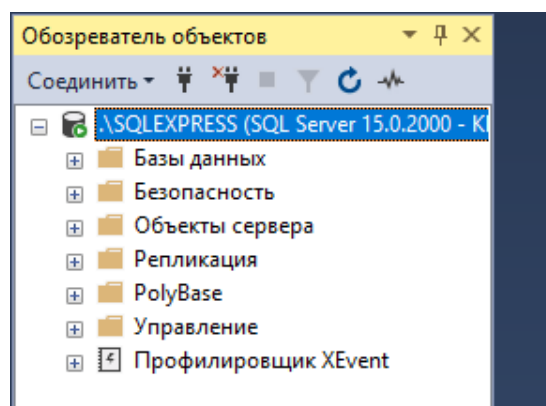


Рисунок 3 - Обозреватель объектов

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

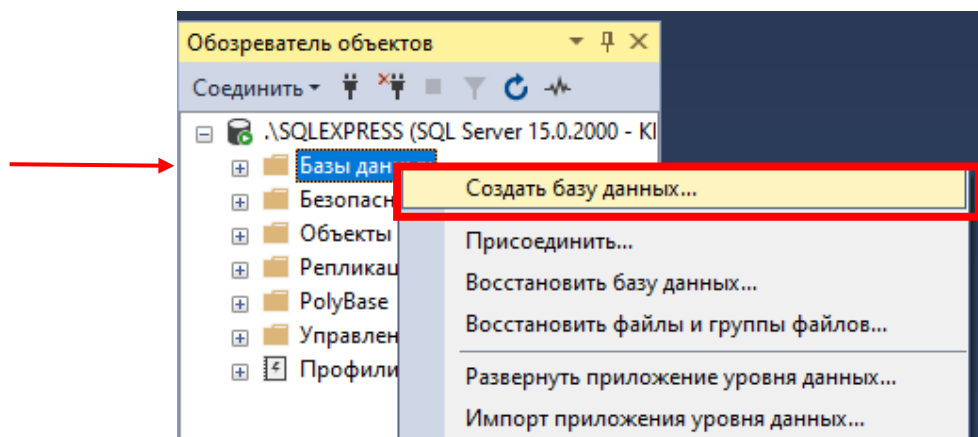


Рисунок 4 - Создание базы данных

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

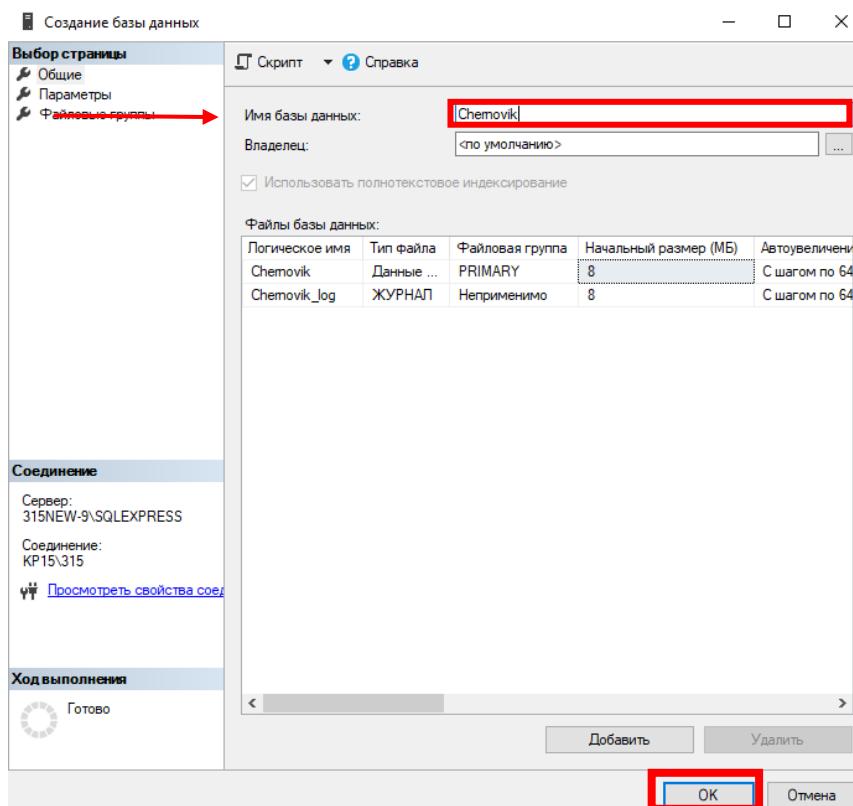


Рисунок 5 - Окно создания базы данных

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

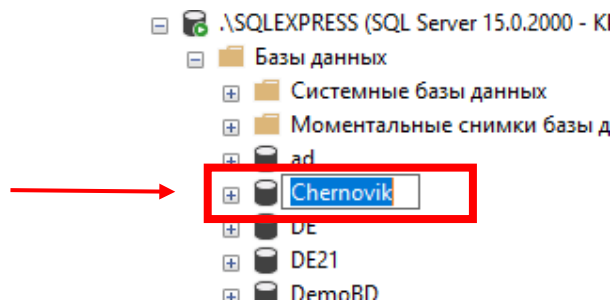


Рисунок 6 - Созданная база данных

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.

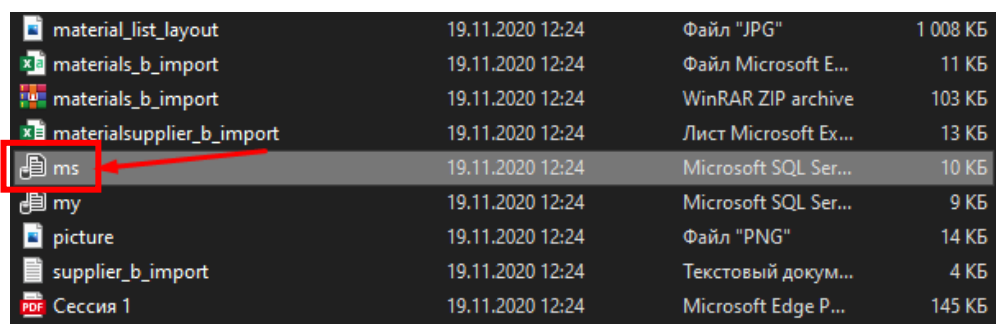


Рисунок 7 - Скрипт базы данных

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)

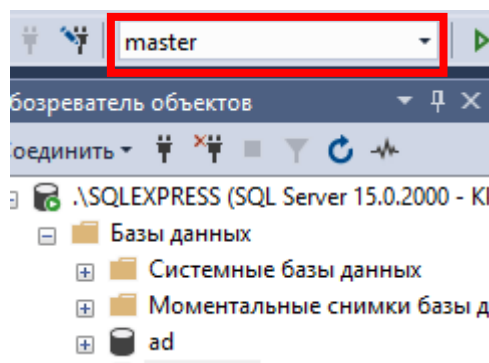


Рисунок 8 - Выбор базы данных

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

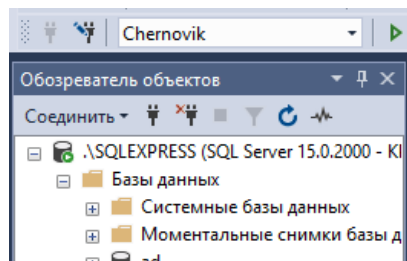


Рисунок 9 - Выбор базы данных

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

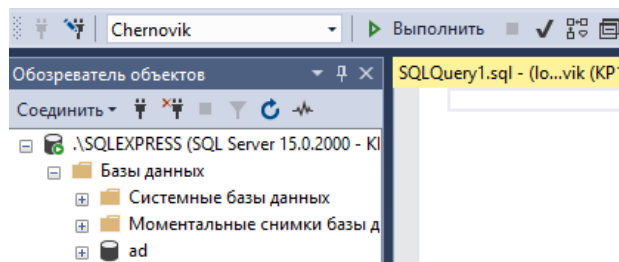


Рисунок 10 - Выполнение скрипта

В результате выполнения данного скрипта в ранее созданной базе данных появятся таблицы (рис.11). Некоторые из данных необходимо будет заполнить с помощью файлов (рис.12): materials\_b\_import, materialsupplier\_b\_import, supplier\_b\_import.

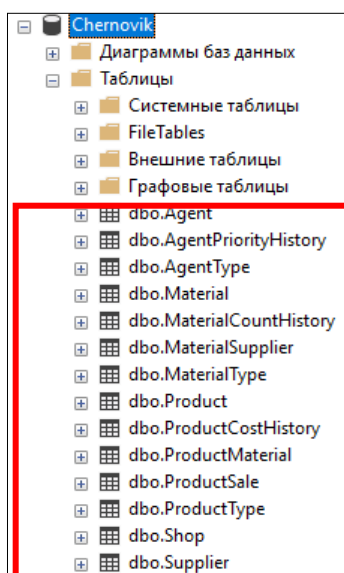


Рисунок 11 - Созданные таблицы



materials	19.11.2020 1:01	Папка с файлами	
material_list_layout	19.11.2020 12:24	Файл "JPG"	1 008 КБ
materials_b_import	28.01.2022 16:19	Файл Microsoft E...	9 КБ
materialsupplier_b_import	19.11.2020 12:24	Лист Microsoft Ex...	13 КБ
ms	19.11.2020 12:24	Microsoft SQL Ser...	10 КБ
my	19.11.2020 12:24	Microsoft SQL Ser...	9 КБ
picture	19.11.2020 12:24	Файл "PNG"	14 КБ
supplier_b_import	19.11.2020 12:24	Текстовый докум...	4 КБ
Сессия 1	19.11.2020 12:24	Microsoft Edge P...	145 КБ

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.

1	Наименование материала	Тип материала	Изображение	Цена	Количество на складе	Минимальное	Количество в	Единица измерения
2	Гранулы зеленый 0х3	Гранулы	не указано	33037.00	На складе: 937	31	10	л
3	Рулон зеленый 0х0	Рулон	\materials\material_25.jpeg	43562 рублей	На складе: 371	32	8	м
4	Нарезка серый 1х1	Нарезка	нет	36682	302	41	7	м
5	Рулон серый 1х0	Рулон	\materials\material_12.jpeg	1998 рублей	811 в наличии	26	10	кг
6	Нарезка белый 1х2	Нарезка	нет	39259	248	27	4	м
7	Нарезка зеленый 3х0	Нарезка	\materials\material_7.jpeg	25985.00	На складе: 345	30	4	кг
8	Спрессованный пак синий 0х2	Пресс	\materials\material_24.jpeg	47443	881 в наличии	14	9	м
9	Спрессованный пак зеленый 2х0	Пресс	нет	25974.00	215 в наличии	25	6	кг
10	Нарезка белый 3х1	Нарезка	отсутствует	27374.00	На складе: 439	50	8	м
11	Гранулы цветной 1х3	Гранулы	\materials\material_11.jpeg	16651.00	20	23	4	кг
12	Рулон синий 1х0	Рулон	нет	49990.00	На складе: 972	40	9	м
13	Спрессованный пак цветной 3х0	Пресс	отсутствует	36740 рублей	На складе: 159	42	5	кг
14	Нарезка белый 0х1	Нарезка	нет	32370.00 Р	На складе: 368	16	10	м
15	Рулон зеленый 1х3	Рулон	нет	12393.00	На складе: 20	17	6	м
16	Спрессованный пак белый 1х2	Пресс	\materials\material_16.jpeg	21359	27 в наличии	28	2	м
17	Спрессованный пак белый 0х2	Пресс	\materials\material_5.jpeg	36343.00	601 в наличии	49	7	м
18	Рулон зеленый 2х2	Рулон	отсутствует	7610.00 Р	94 в наличии	6	4	кг
19	Спрессованный пак зеленый 1х2	Пресс	\materials\material_3.jpeg	35304.00	1	12	4	м
20	Рулон зеленый 1х1	Рулон	отсутствует	28235 руб.	884	5	1	кг

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)

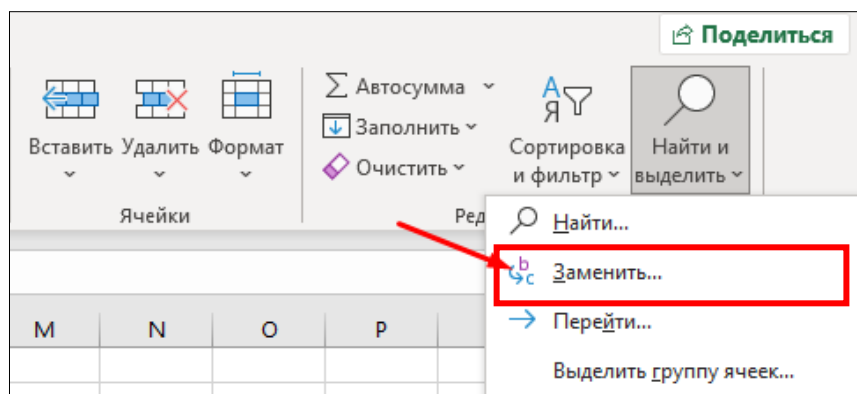


Рисунок 14 - Кнопка "Заменить..."

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



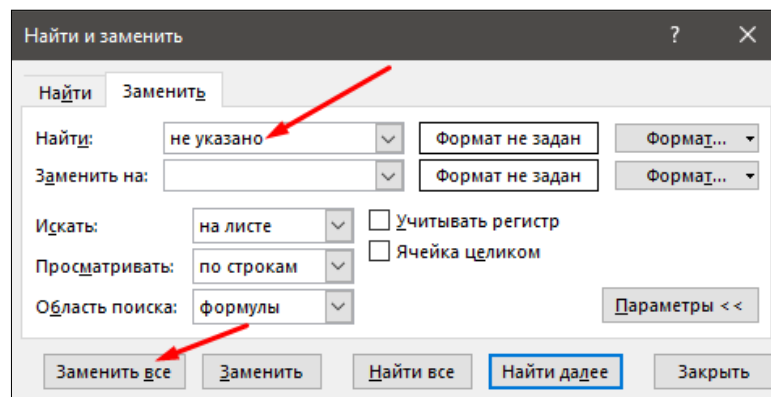


Рисунок 15 - Окно "Найти и заменить"

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

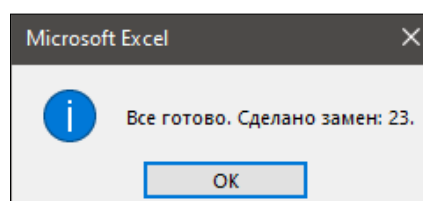


Рисунок 16 - Окно информации

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

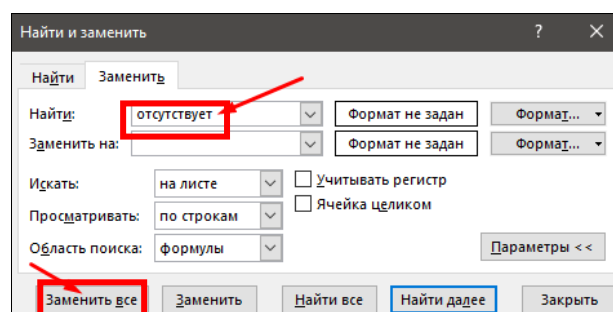


Рисунок 17 - Найти "отсутствует"

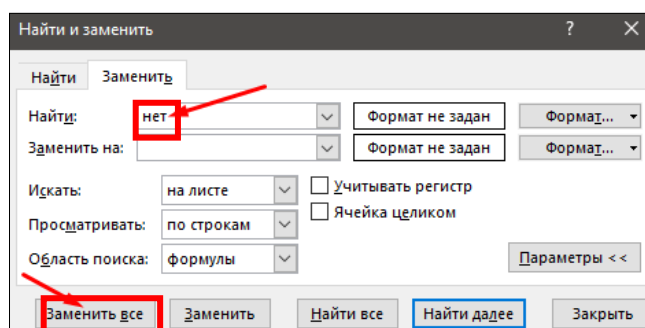


Рисунок 18 - Найти "нет"

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

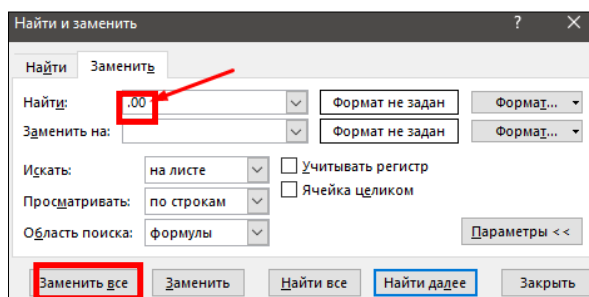


Рисунок 19 - Найти ".00"

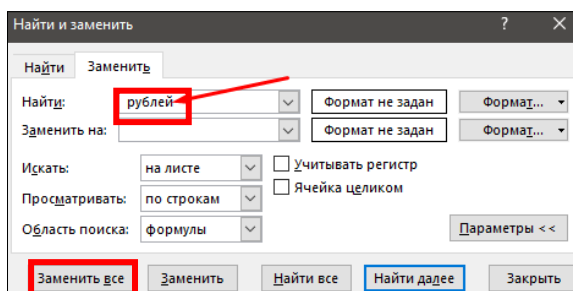


Рисунок 20 - Найти "рублей"

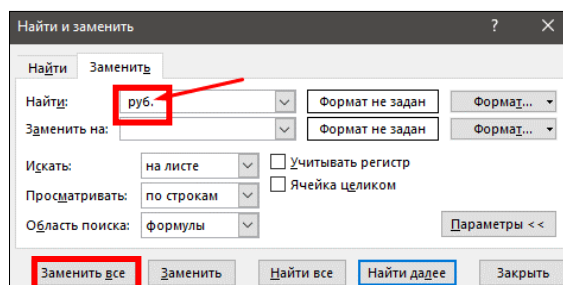


Рисунок 21 - Найти "руб."

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

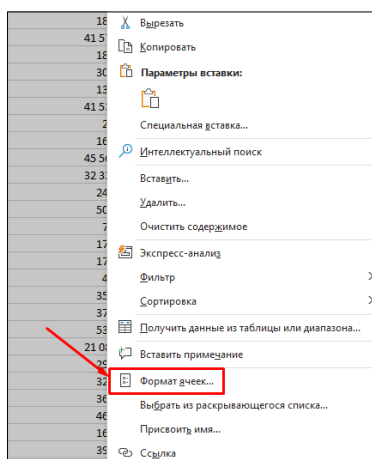


Рисунок 22 - Формат ячеек

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

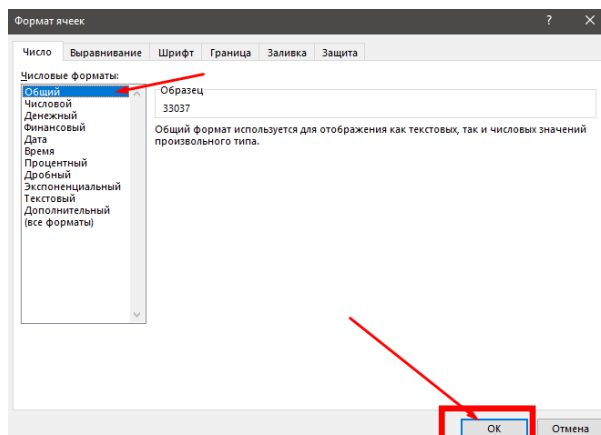


Рисунок 23 - Окно "Формат ячеек"

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

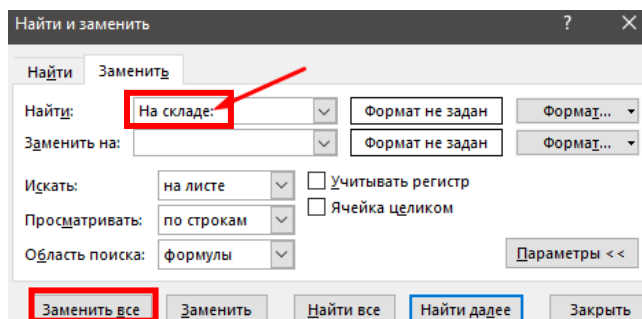


Рисунок 24 - Найти "На складе: "

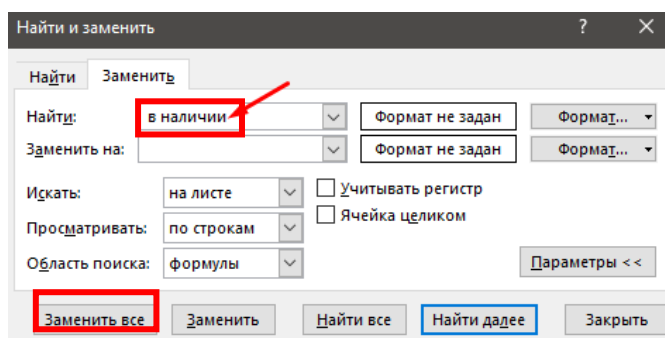


Рисунок 25 - Найти "в наличии"

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



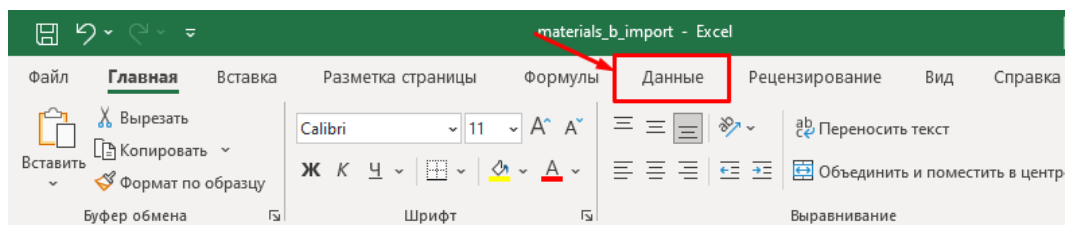


Рисунок 26 - Вкладка "Данные"

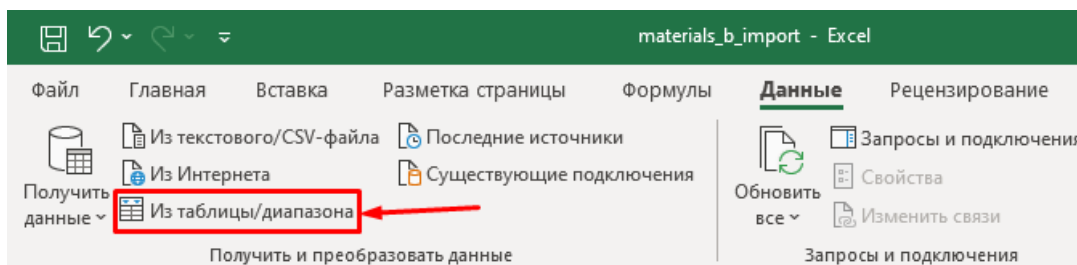


Рисунок 27 - "Из таблицы/диапазона"

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

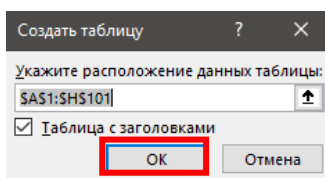


Рисунок 28 - Расположение таблицы

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)

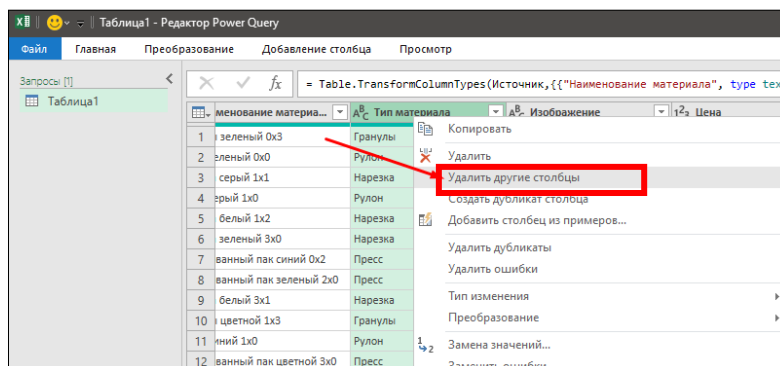


Рисунок 29 - Удаление других столбцов

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

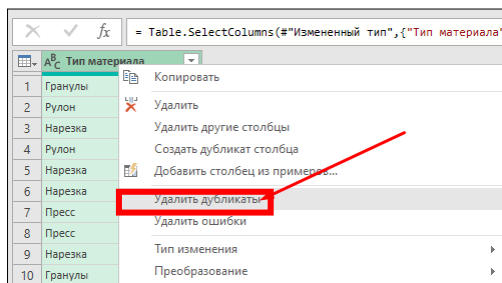


Рисунок 30 - Удаление дубликатов в столбце

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

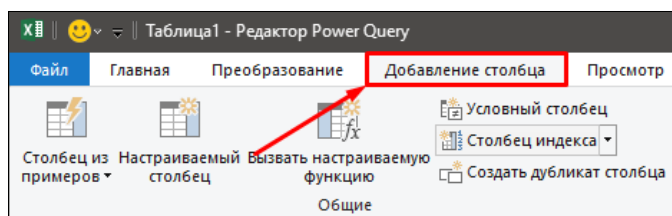


Рисунок 31 - Добавление столбца индекса

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

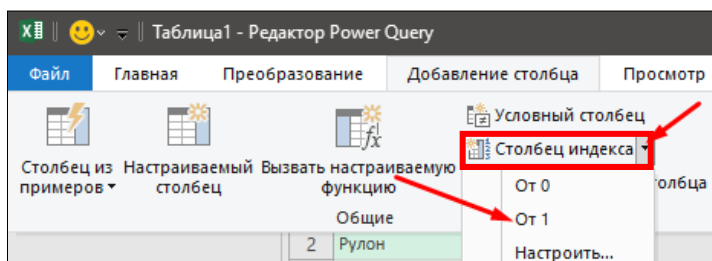


Рисунок 32 - Столбец индекса от 1

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

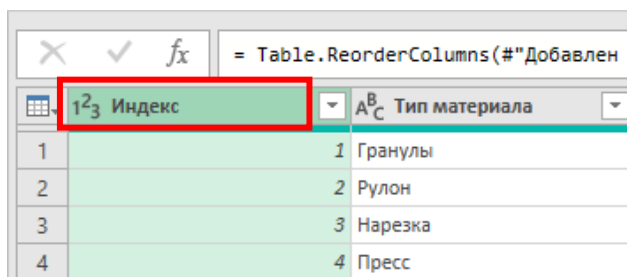


Рисунок 33 - Столбец индекса

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

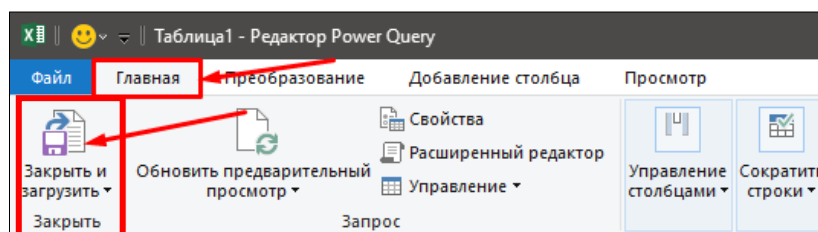


Рисунок 34 - Сохранение таблицы

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.

1	Индекс	Тип материала
2	1	Гранулы
3	2	Рулон
4	3	Нарезка
5	4	Пресс

Рисунок 35 - Созданная таблица

Индекс	Тип материала	Столбе
1	Гранулы	0
2	Рулон	0
3	Нарезка	0
4	Пресс	0

Рисунок 36 - Столбец с нулями

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

1	Гранулы	0
2	Рулон	0
3	Нарезка	0
4	Пресс	0

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)

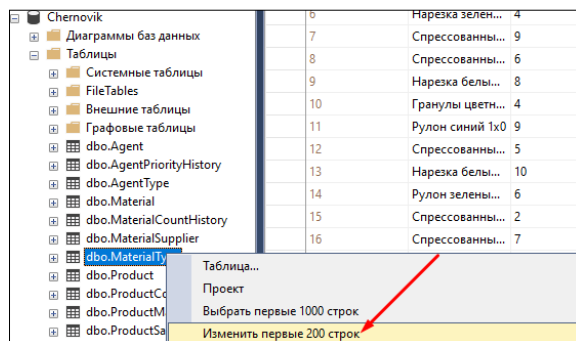


Рисунок 38 - Изменение таблицы "MaterialType"

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»)

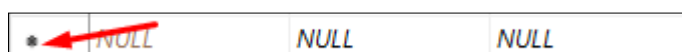


Рисунок 39 - Выделение строк

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)

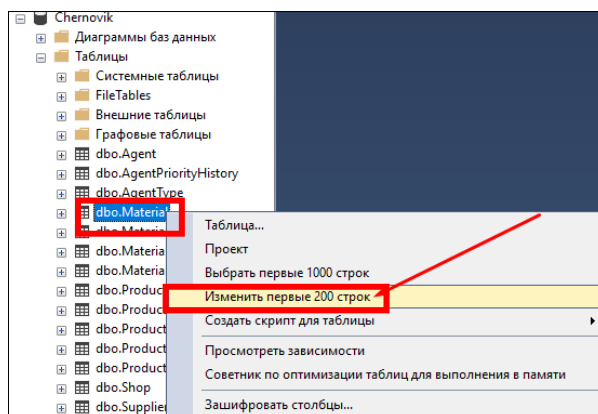


Рисунок 40 - Изменить первые 200 строк

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

ID	Title	CountInPack	Unit	CountInStock	MinCount	Description	Cost	Image	MaterialTypeID
----	-------	-------------	------	--------------	----------	-------------	------	-------	----------------

Рисунок 41 - Расположение столбцов

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

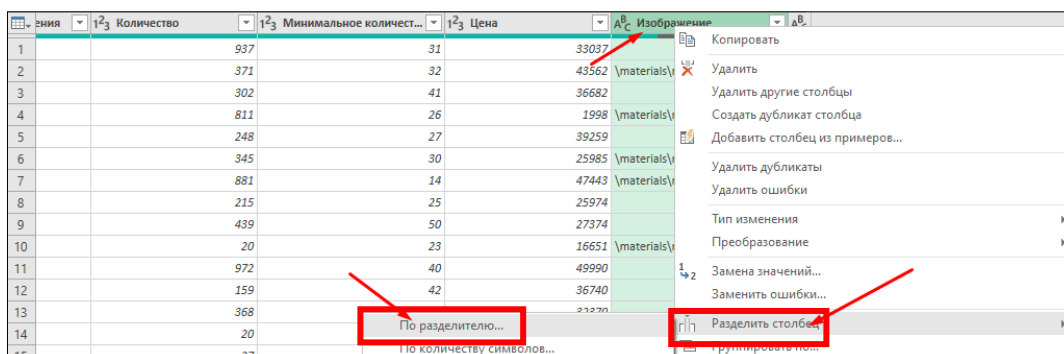


Рисунок 42 - Разделение столбца

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

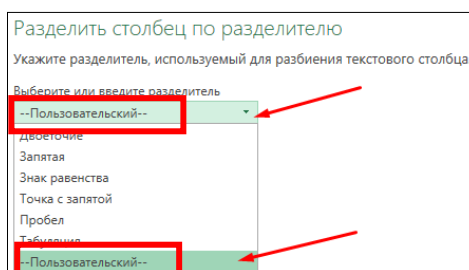


Рисунок 43 - Разделитель "Пользовательский"

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

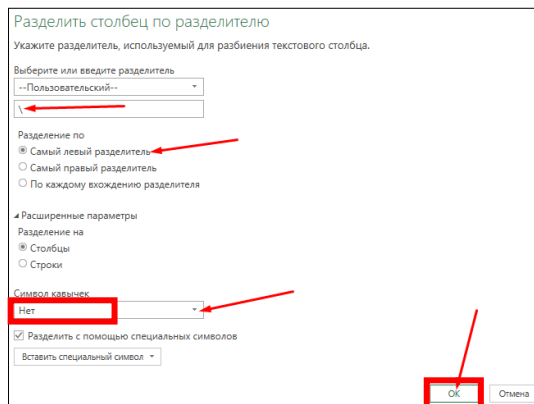


Рисунок 44 – Окно разделения столбца

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)

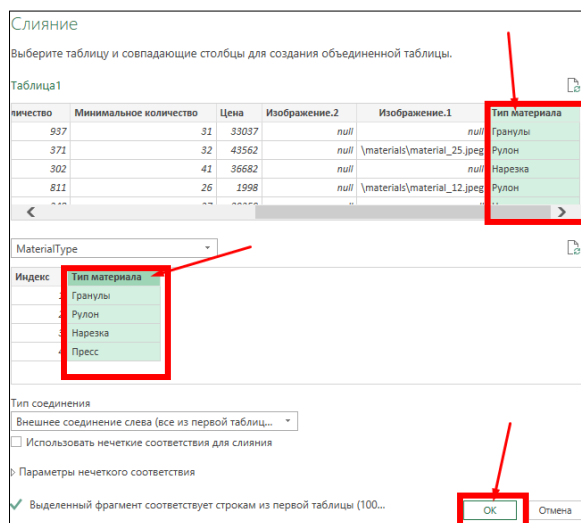


Рисунок 46 - Окно слияния

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

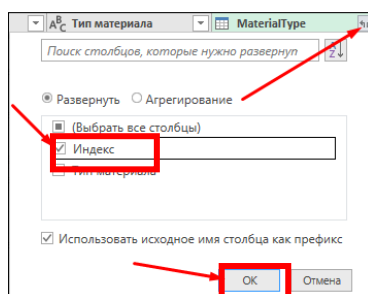


Рисунок 47 - Настройка столбца

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

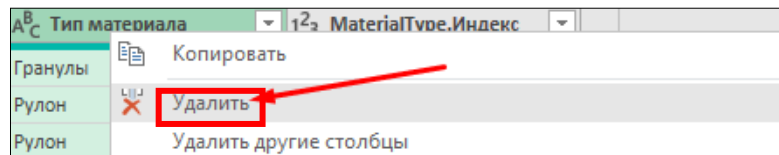


Рисунок 48 - Удаление столбца

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

Копируем записи таблицы (либо нажимаем комбинацию «ctrl + с», 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.

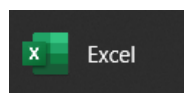


Рисунок 49 - Приложение Excel

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

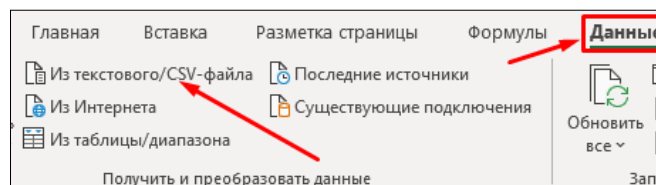


Рисунок 50 - Получение данных из текстового файла

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

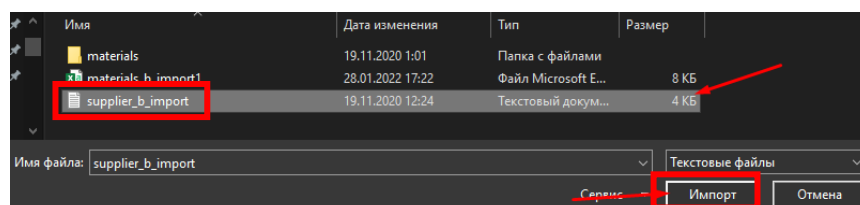


Рисунок 51 - Выбор файла

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

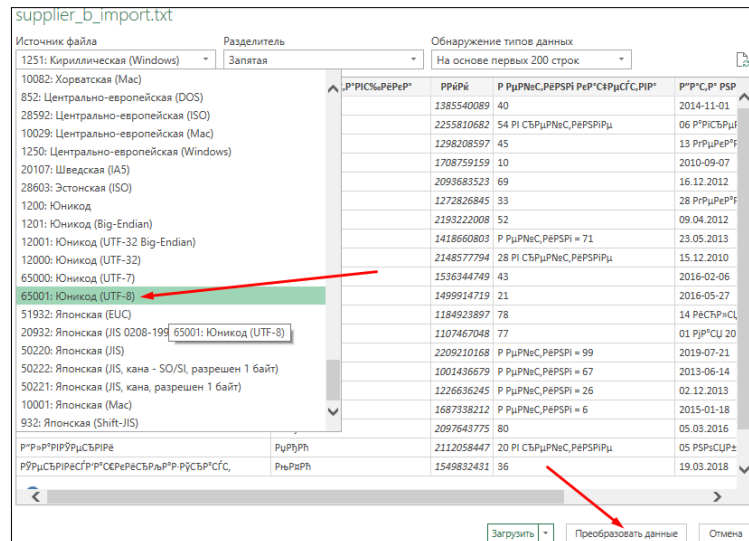


Рисунок 52 - Преобразование данных

Сохраняем таблицу с помощью кнопки «Закреть 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».

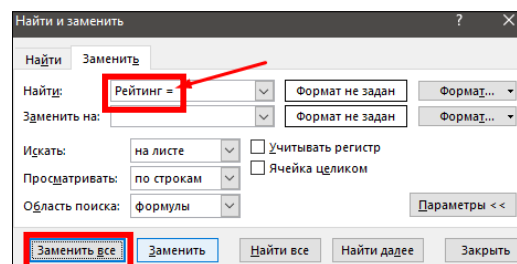


Рисунок 53 - Найти "Рейтинг ="

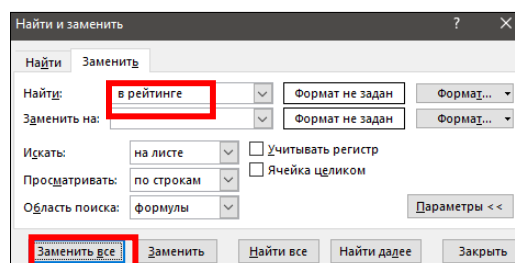




Рисунок 54 - Найти "в рейтинге"

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)

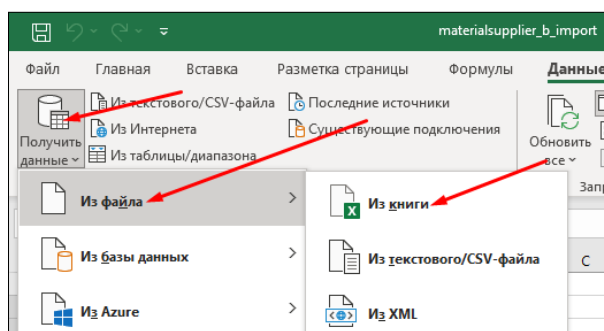


Рисунок 55 - Открытие других книг

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»

Слияние

Выберите таблицу и совпадающие столбцы для создания объединенной таблицы.

Таблица1\_2

Наименование материала	Возможный поставщик
Спрессованный лак белый 1x1	ОрионБухМетал
Спрессованный лак белый 1x1	ОблВектор
Нарезка цветной 2x2	МонтажНефтьОрион
Спрессованный лак белый 1x1	ХозБаширГла
Нарезка цветной 2x2	Компания МорКаза

materials\_b\_import1

Индекс	Наименование материала	Количество в упаковке	Единица измерения	Количество на складе
1	Гранулы зеленый 0x3	10	л	937
2	Рулон зеленый 0x0	8	м	371
4	Рулон серый 1x0	10	кг	811
3	Нарезка серый 1x1	7	м	302

Тип соединения

Внешнее соединение слева (все из первой таблиц...)

☐ Использовать нечеткие соответствия для слияния

Параметры нечеткого соответствия

✓ Выделенный фрагмент соответствует строкам из первой таблицы (100...)

OK Отмена

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»

Слияние

Выберите таблицу и совпадающие столбцы для создания объединенной таблицы.

Таблица1\_2

materials_b_import1.Индекс	Возможный поставщик
96	ОрионБухМетал
96	ОблВектор
96	ХозБаширГла
2	ОблВектор
2	Компания ТелекомРосКрепПром

supplier\_b\_import1

Индекс	Наименование поставщика	ИНН	Дата начала работы с поставщиком	Рейтинг качества	T
	ОблВектор	1385540089	01.11.2014	40	N
	Строй	2255810682	06.04.2010	54	C
	МонтажНефтьОрион	1298208597	13.12.2014	45	C
	ХозБаширГла	1708759159	07.09.2010	10	C

Тип соединения

Внешнее соединение слева (все из первой таблиц...)

☒ Использовать нечеткие соответствия для слияния

Параметры нечеткого соответствия

✓ Выделенный фрагмент соответствует строкам из первой таблицы (100...)

OK Отмена

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)

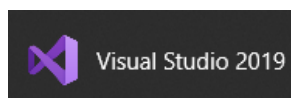


Рисунок 58 - Приложение Visual Studio

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».

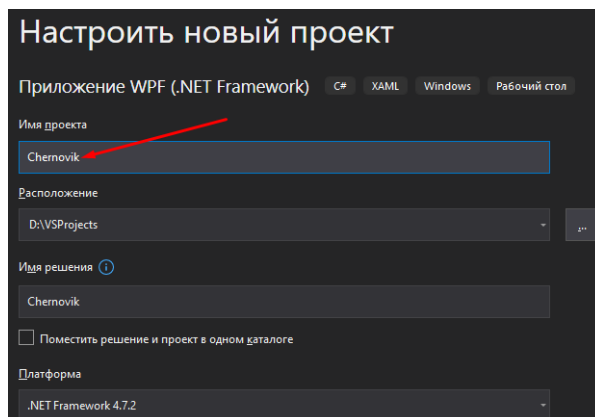


Рисунок 60 - Настройка проекта

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

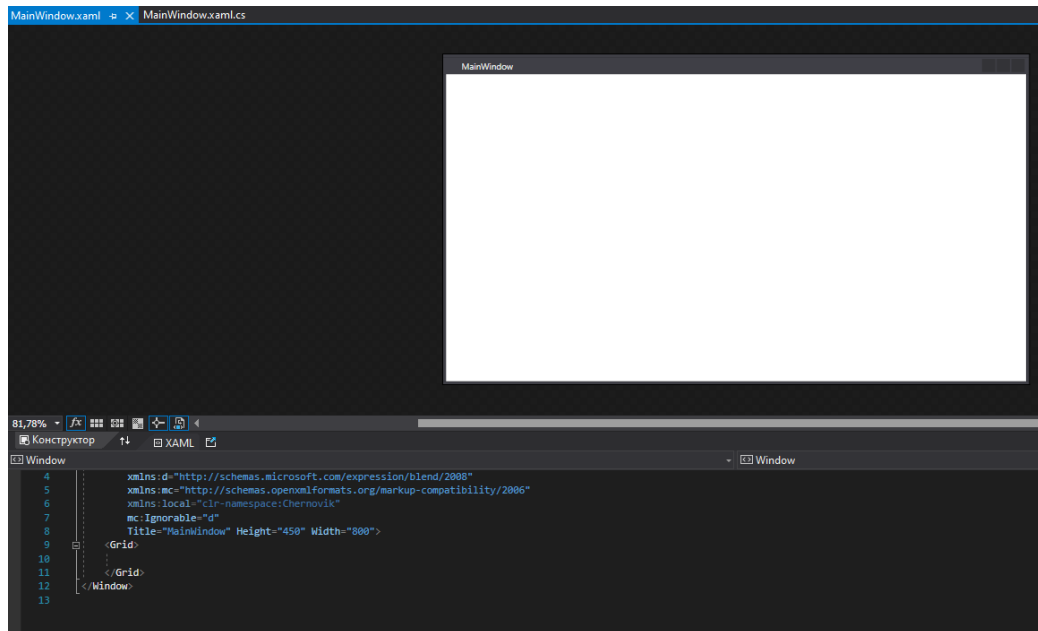


Рисунок 61 - Внешний вид программы

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)

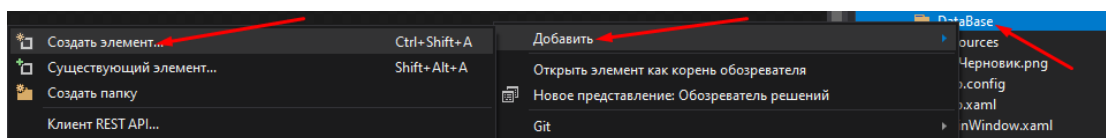


Рисунок 62 - Добавление модели

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

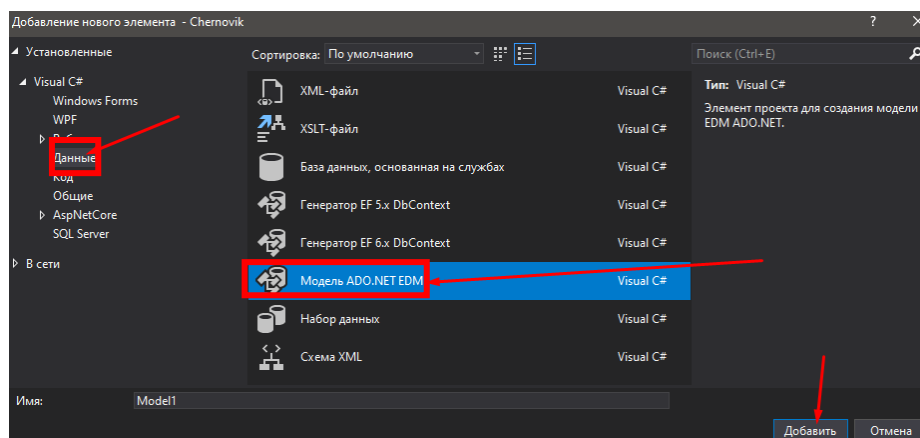


Рисунок 63 - Добавление Модели ADO.NET EDM

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



Какое подключение к данным будет использоваться приложением для подключения к базе данных?

elena-pc\sqlexpress.Chernovik.dbo Создать соединение...

Возможно, эта строка подключения содержит конфиденциальные данные (например, пароль), которые требуются для подключения к базе данных. Хранение конфиденциальных данных в строке подключения может представлять угрозу безопасности. Включить конфиденциальные данные в строку подключения?

☐ Нет, исключить конфиденциальные данные из строки подключения. Они будут заданы в коде приложения.

☐ Да, включить конфиденциальные данные в строку подключения.

Строка подключения:

```
metadata=res://*/DataBase.ModelEF.csdl|res://*/DataBase.ModelEF.ssdl
res://*/DataBase.ModelEF.msl;provider=System.Data.SqlClient;provider connection string="data source=.\sqlexpress;initial catalog=Chernovik;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework"
```

☒ Сохранить параметры соединения в App.Config как:

ChernovikEntities

< Назад Далее > Готово Отмена

Рисунок 64 - Создание соединения

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

Свойства подключения

Введите данные для подключения к выбранному источнику данных или нажмите кнопку "Изменить", чтобы выбрать другой источник данных и (или) поставщик.

Источник данных: Microsoft SQL Server (SqlClient) Изменить...

Имя сервера: .\sqlexpress Обновить

Вход на сервер

Проверка подлинности: Проверка подлинности Windows

Имя пользователя:

Пароль:

☐ Сохранить пароль

Подключение к базе данных

☒ Выберите или введите имя базы данных: Chernovik

☐ Прикрепить файл базы данных: Обзор...

Логическое имя:

Дополнительно...

Проверить подключение ОК Отмена

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)



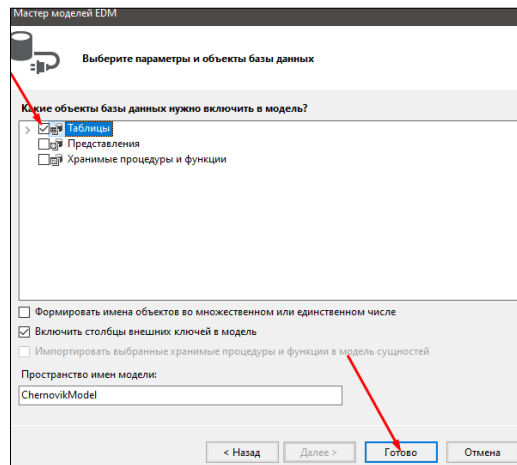


Рисунок 66 - Выбор таблиц

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

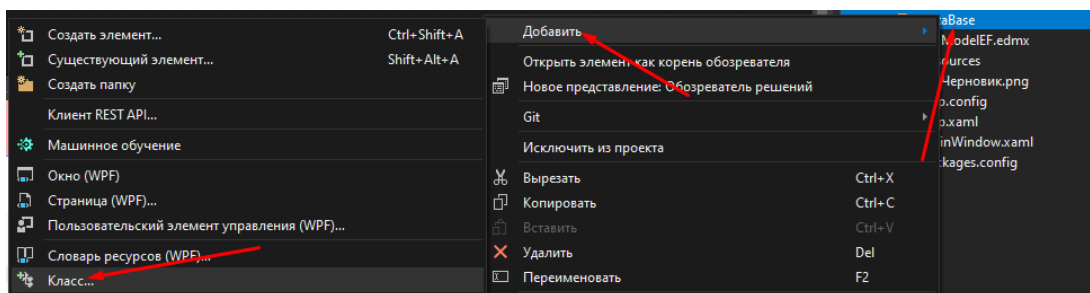


Рисунок 67 - Создание класса

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 дизайнера 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 находиться 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>

    <ScrollView>
```

```

<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>
</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();
```