# **Тема 1 Работа в MS Excel, как в реляционные базы данных.**

# Разработка простейшей базы данной в программе MS Excel



Рисунок 1. Трудоустройство выпускников

## 1.2 Функции ВПР и ГПР, операции с диапазонами

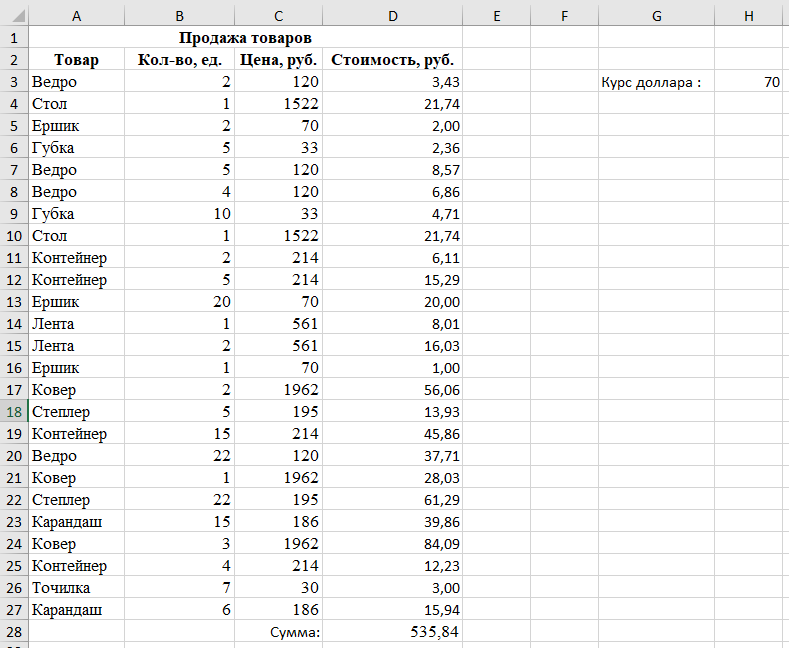


Рисунок 2. Продажа товаров

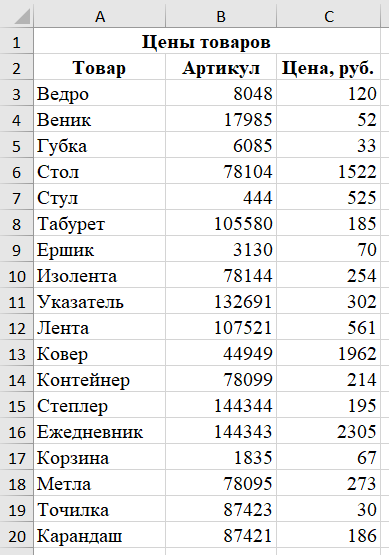


Рисунок 3. Цены товаров

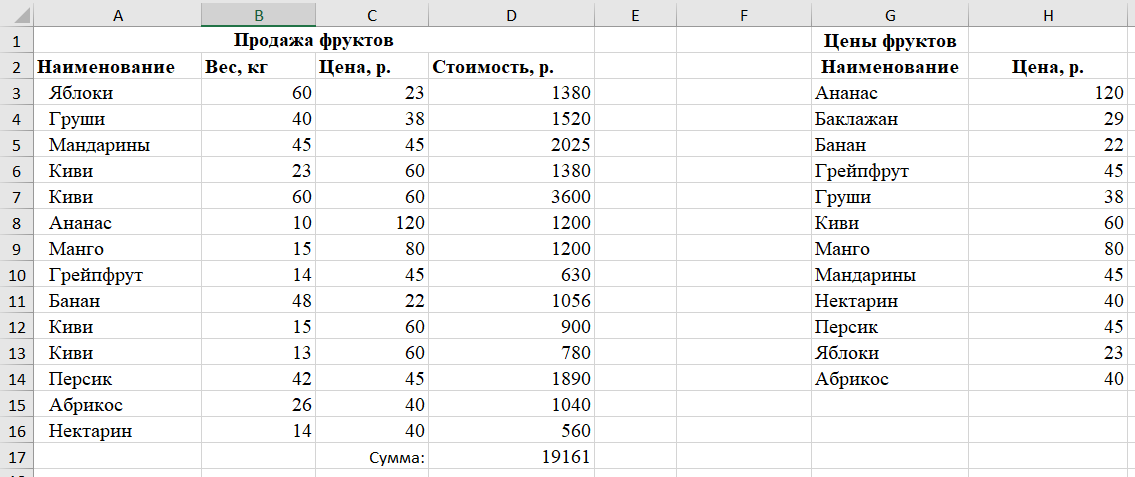


Рисунок 4. Продажа фруктов

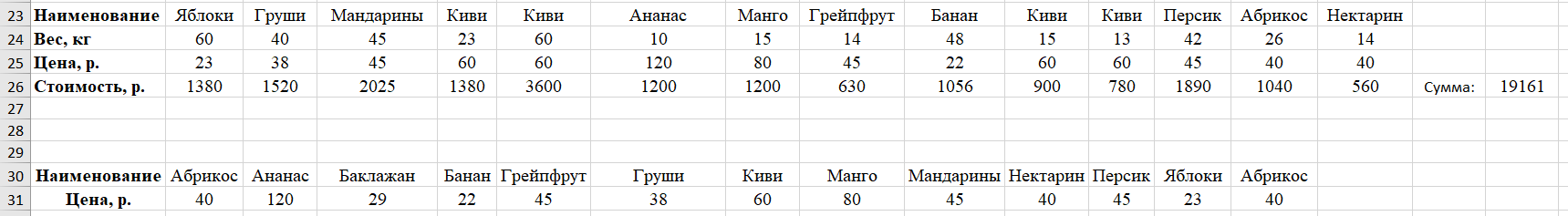


Рисунок 5. Цены фруктов

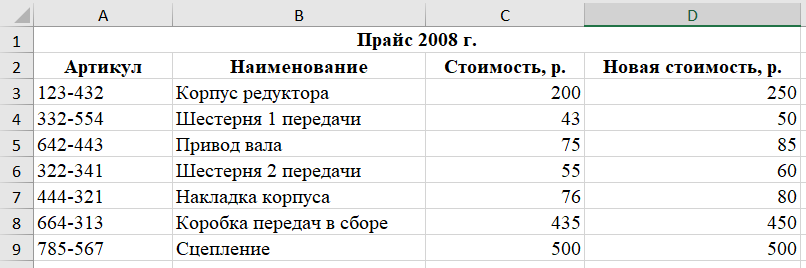


Рисунок 6. Прайс 2008 года

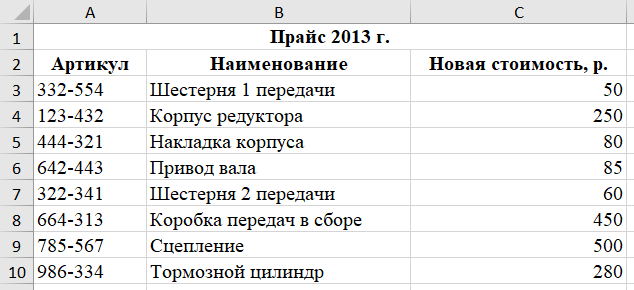


Рисунок 7. Прайс 2013 года

# **Тема 2. Руководство пользователя по работе с GitHub**

1. Проверить наличие программы на компьютере



Рисунок 8. Git

1. Для выгрузки файла в Git нужно перейти по данной ссылке: <https://try.gogs.io/user/login>. Либо в поисковике вбиваем Gogs.
2. Откроется Gogs в котором необходимо зарегистрироваться. (В поле Имя пользователя пишем: Фамилия Имя на английском) Сообщение, которое должно прийти на почту при регистрации, может попасть в спам.
3. После регистрации открывается Панель управления в которой нужно добавить репозиторий

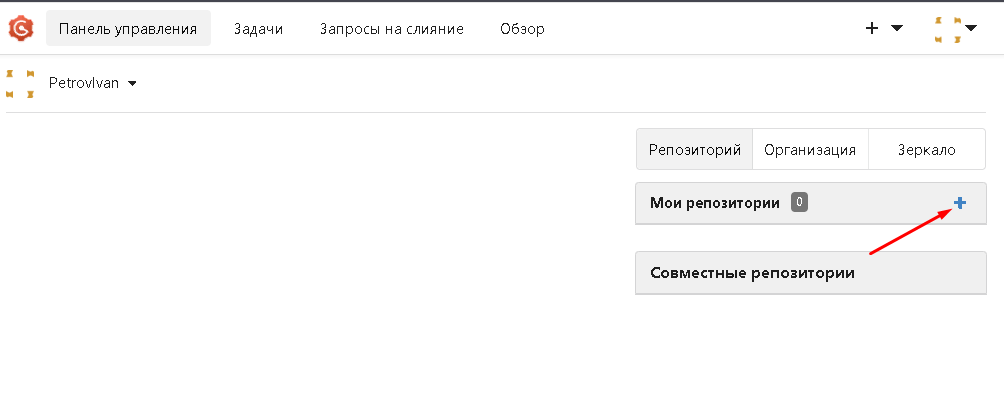


Рисунок 9. Панель управления

1. Имя репозитория First. Видимость ставим галочку Личный репозиторий. Описание обязательно прописываем, например: «Первая программа».

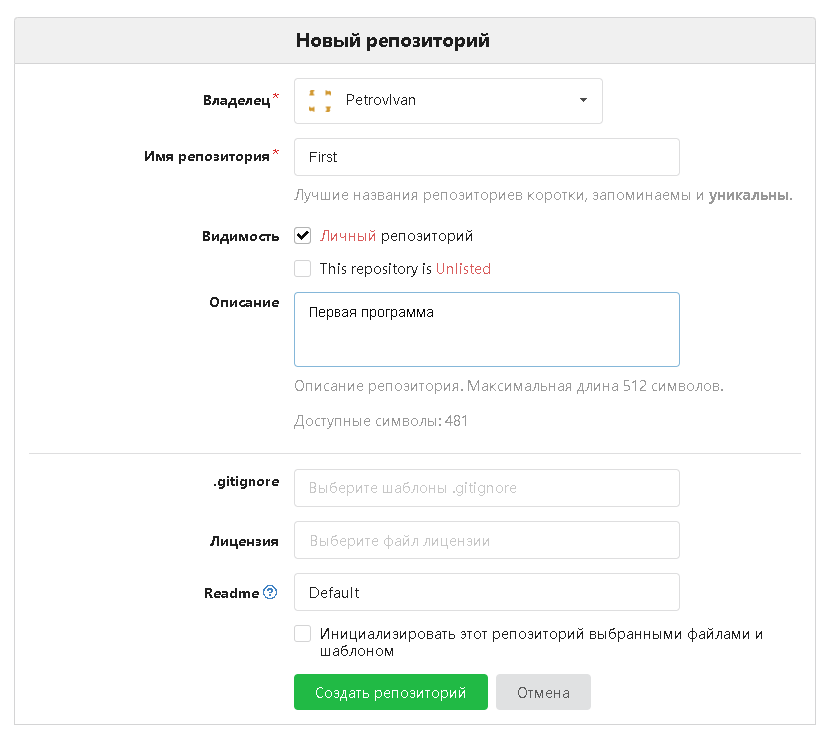


Рисунок 10. Новый репозиторий

1. Появляется краткое руководство, которое позже нам понадобится.
2. Открываем Git Bash Here



Рисунок 11. Git Bash Here

1. Открывается командная строка

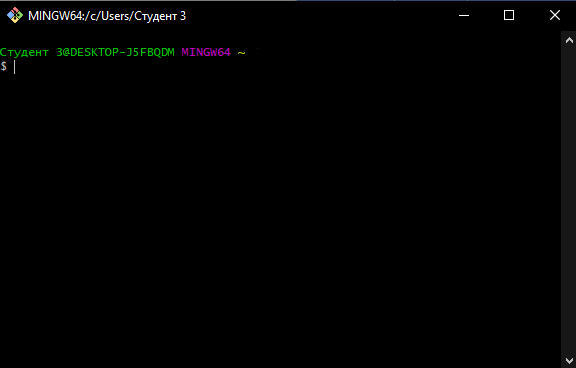


Рисунок 12. Командная строка

1. Используем Краткое руководство из Gogs перестраивая под себя

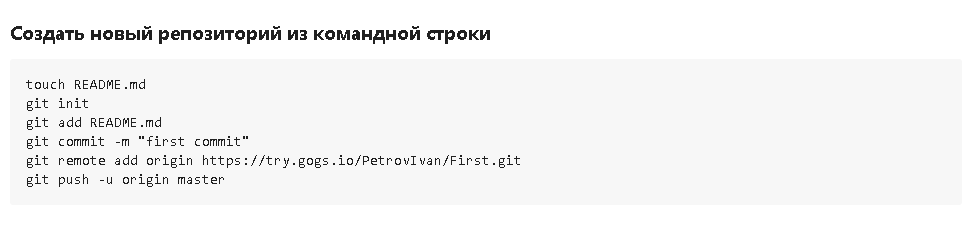


Рисунок 13. Новый репозиторий из командной строки

1. В строке git remote add origin… ссылка отличается в зависимости от аккаунта
2. Все что подчеркнуто красной линии написано у вас в Кратком руководстве в Gogs.

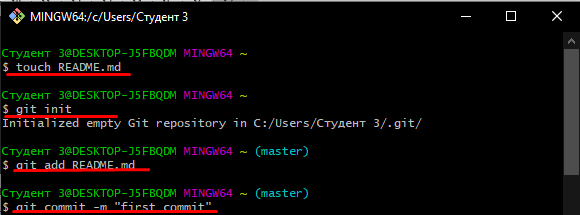


Рисунок 14. Создание репозитория

1. Если появляется данная ошибка

То нужно прописать следующие строчки

Git config –global user.email “электронная почта на которую зарегистрирован Gogs”

Git config –global user.Name «свое имя»

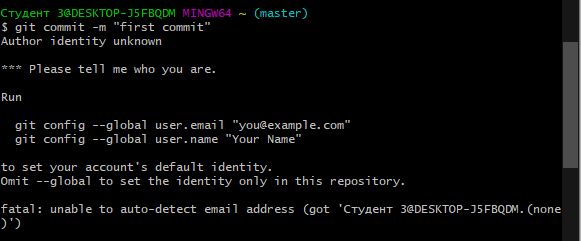


Рисунок 15. Авторизация

1. Должно получиться следующее

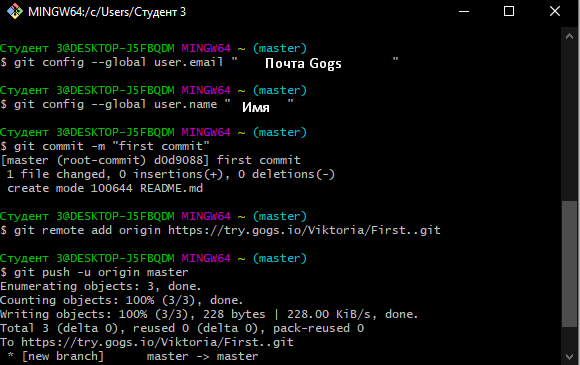


Рисунок 16. Выгрузка в ветвь master

1. Если сделать все верно, то появится Первый репозиторий

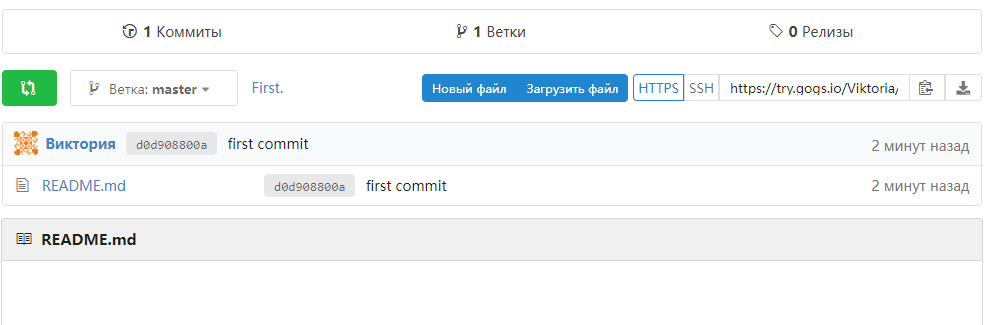


Рисунок 17. Первый репозиторий

1. Создадим отдельную ветку

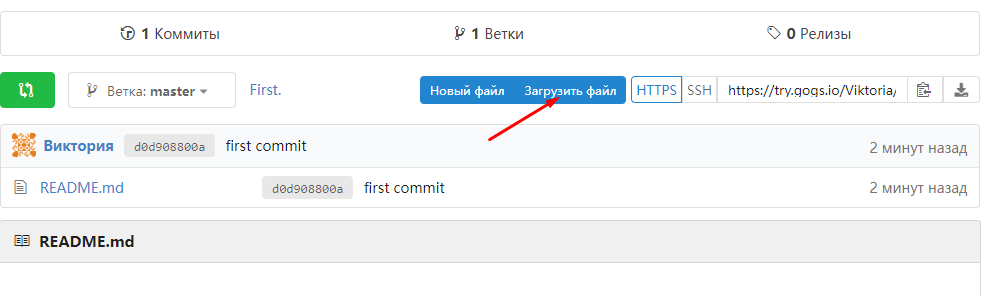


Рисунок 18. Создание ветки

1. Добавляем файл, выбираем пункт Создать новую ветвь и называем.

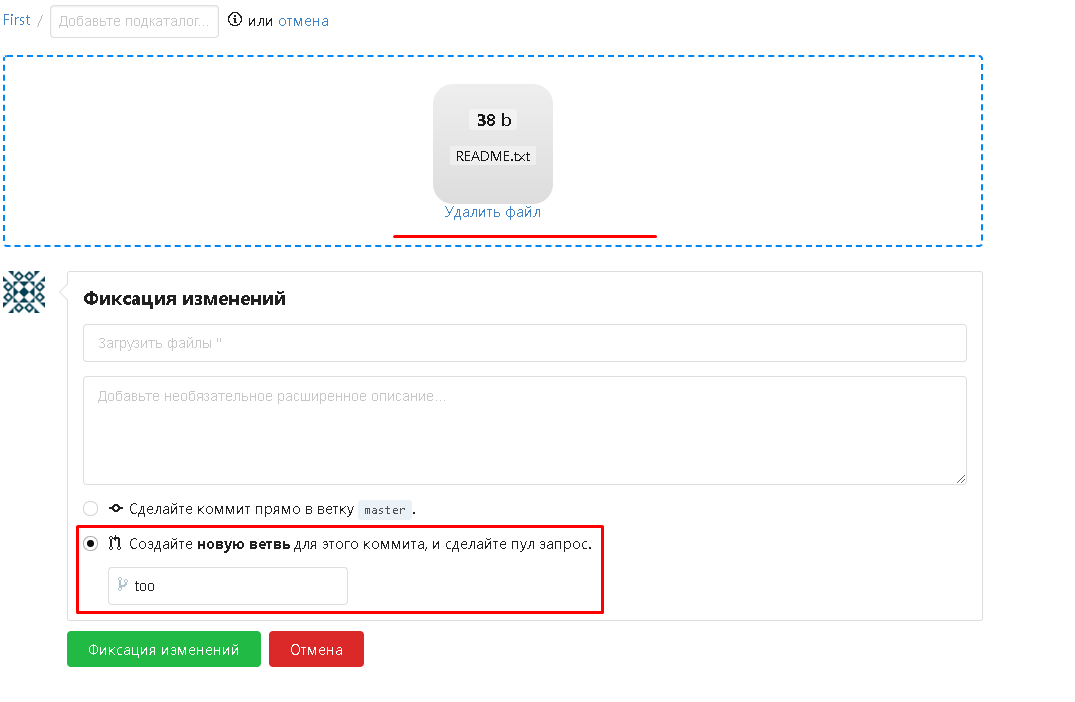


Рисунок 19. Ветвь too

1. Добавляем описание и нажимаем Фиксация изменений
2. Возвращаемся в репозиторий и видим, что новая ветка добавлена

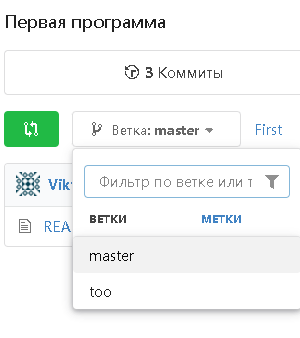


Рисунок 20. Ветки

**Тема 3 Разработка базы данных**

## 3.1 Моделирование данных. ERD

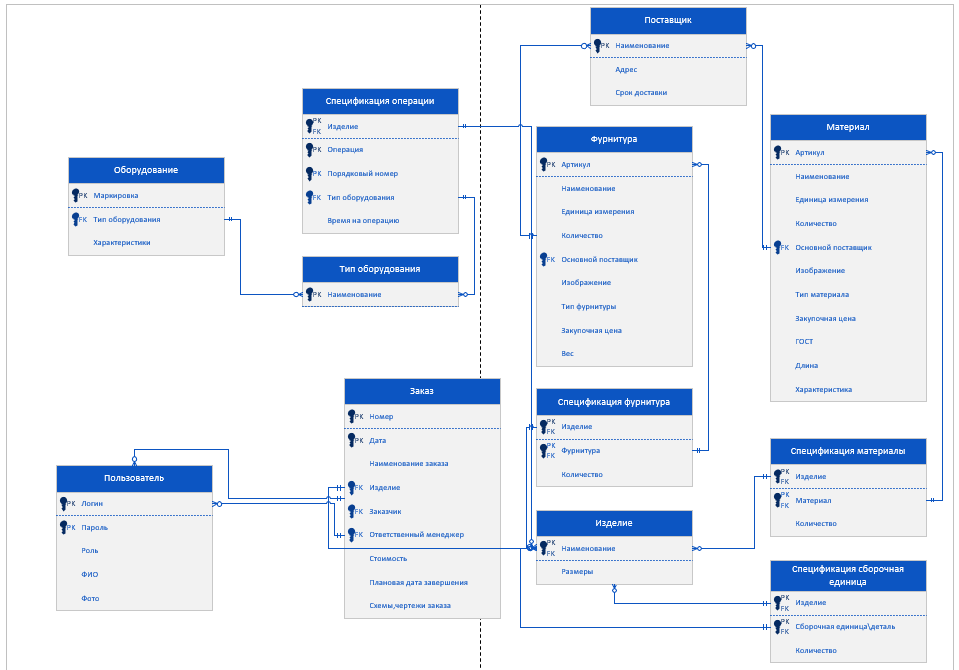


Рисунок 21. ERD Visio

## 3.2 Создание базы данных по средствам запросов

USE DE\_Anoshina\_1

CREATE TABLE Studenty(

Familiya Varchar(50) PRIMARY KEY,

Imya Varchar(50) NOT NULL,

Otchestvo Varchar(50) NOT NULL,

Vozrast INT NOT NULL

)

CREATE TABLE Predmety(

Nazvanie Varchar(50) PRIMARY KEY,

Kabinet INT NOT NULL,

Prepodavately Varchar(50) NOT NULL

)

CREATE TABLE Kabinety(

Kabinet INT PRIMARY KEY,

Korpus INT NOT NULL

)

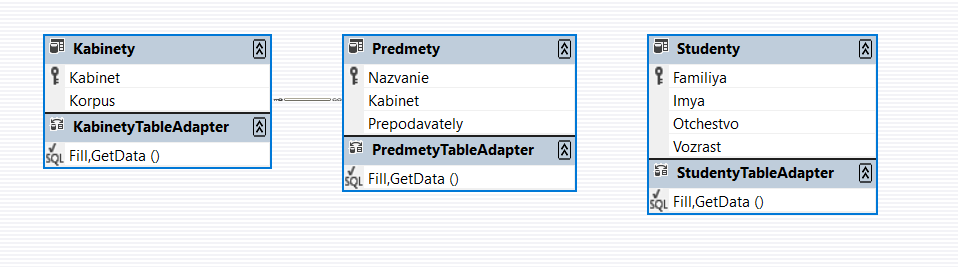


Рисунок 22. Таблицы

## 3.3 Создание аналитических запросов

# **Тема 4 Разработка приложения для работы в день С-1 (руководство пользователя)**

## 4.1 База данных

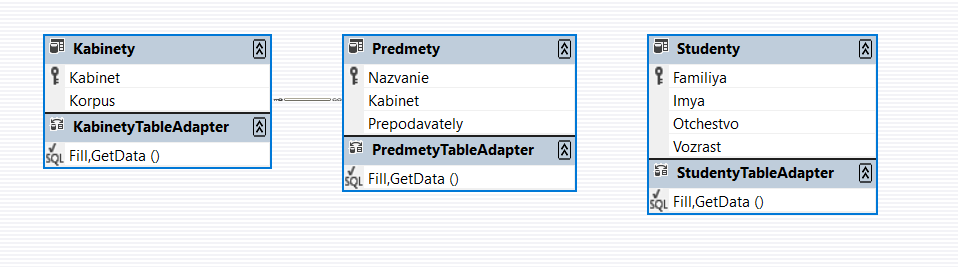


Рисунок 23. База данных

## 4.2 Разработка форм авторизации и поиска

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace DE\_Anoshina\_1

{

public partial class Avtorizacziya : Form

{

public Avtorizacziya()

{

InitializeComponent();

}

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

{

this.Close(); //Закрыть форму

}

private void btnAvtorizacziya\_Click(object sender, EventArgs e) // Авторизация

{

SqlConnection sqlConnection = new SqlConnection(@"Data Source=DESKTOP-J5FBQDM\SQLEXPRESS;Initial Catalog=DE\_Anoshina\_1;Integrated Security=True");

string query = "SELECT \* FROM [Studenty] WHERE [Familiya] = '" + tbFamiliya.Text.Trim() + "' AND [Imya] = '" + tbImya.Text.Trim() + "'";

SqlDataAdapter sda = new SqlDataAdapter(query, sqlConnection);

DataTable dtbl = new DataTable();

sda.Fill(dtbl);

if (dtbl.Rows.Count == 1) //проверка наличия данных

{

MenuPol menu = new MenuPol();

menu.Show();

this.Hide();

}

else

{

MessageBox.Show("Неверная Фамилия или Имя");

tbFamiliya.Clear();

tbImya.Clear();

}

}

}

}

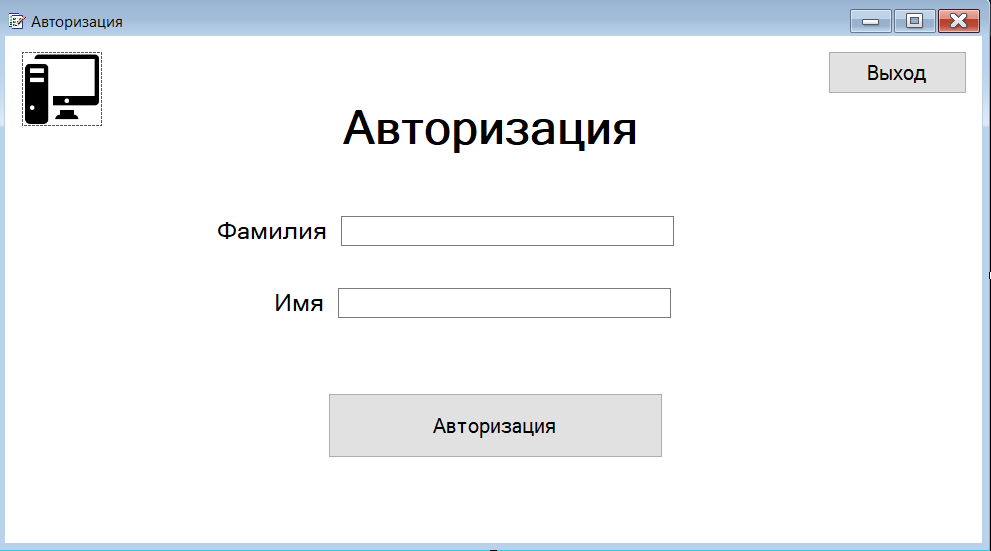


Рисунок 24. Форма Авторизация

## 4.3 Разработка форм фильтрации и сортировки

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace DE\_Anoshina\_1

{

public partial class MenuPol : Form

{

public MenuPol()

{

InitializeComponent();

}

private void btnVihod\_Click(object sender, EventArgs e) //Возвращение на предыдущую форму

{

Avtorizacziya avt = new Avtorizacziya();

avt.Show();

this.Hide();

}

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

{

this.Validate();

this.predmetyBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.dE\_Anoshina\_1DataSet);

}

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

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "dE\_Anoshina\_1DataSet.Predmety". При необходимости она может быть перемещена или удалена.

this.predmetyTableAdapter.Fill(this.dE\_Anoshina\_1DataSet.Predmety);

}

private System.Windows.Forms.DataGridViewColumn COL;

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

{

COL = new System.Windows.Forms.DataGridViewColumn();

switch (lbSortirovka.SelectedIndex)

{

case 0:

COL = dataGridViewTextBoxColumn1;

break;

case 1:

COL = dataGridViewTextBoxColumn2;

break;

case 2:

COL = dataGridViewTextBoxColumn3;

break;

}

if (cbVozrastanie.Checked)

predmetyDataGridView.Sort(COL,

System.ComponentModel.ListSortDirection.Ascending); //Сортировка по возрастанию

else

predmetyDataGridView.Sort(COL,

System.ComponentModel.ListSortDirection.Descending); //Сортировка по убыванию

}

}

}

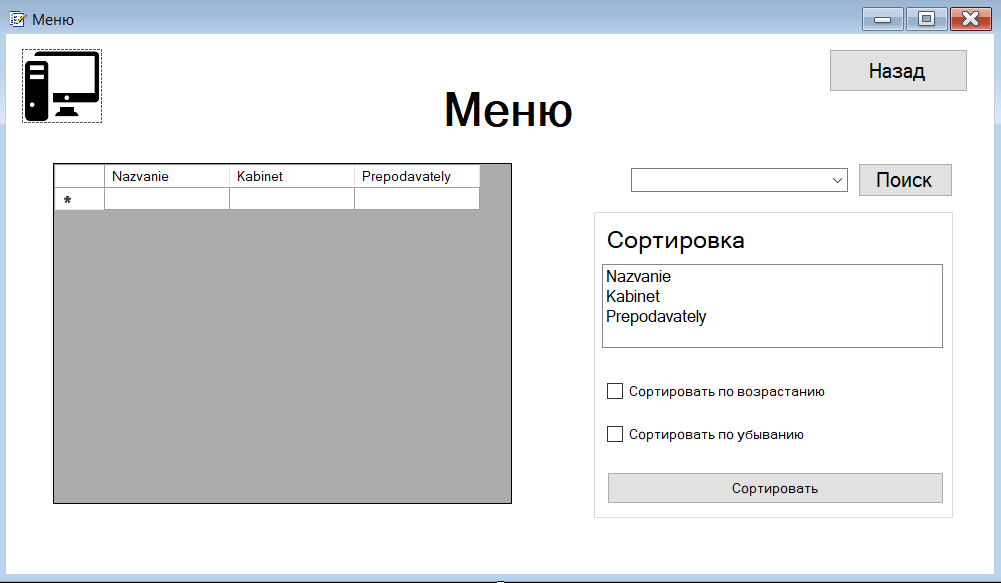


Рисунок 25. Форма фильтрации и сортировки

# **Тема 5 Разработка desktop приложений**

## 5.2 Разработка ERD диаграмм

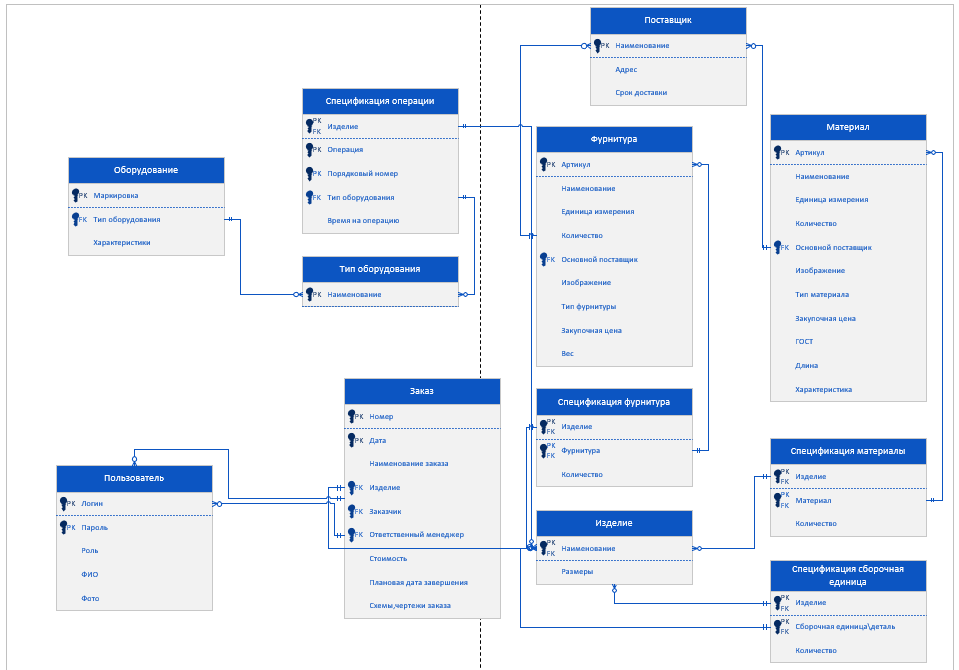


Рисунок 25. ERD Visio

## 5.2.1 Работа в MS Visio

## 5.2.2 Подготовка данных для импорта

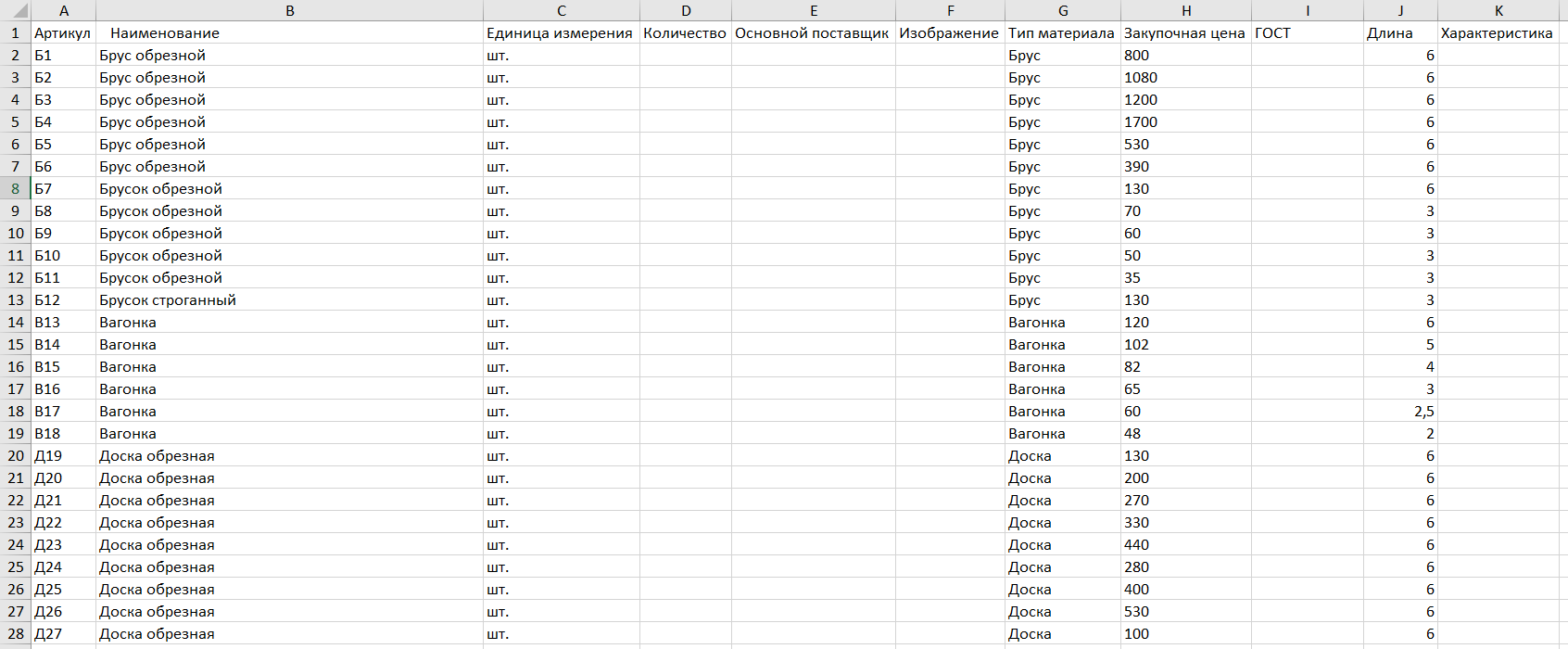


Рисунок 26. Готовая таблица для импорта «Материалы»

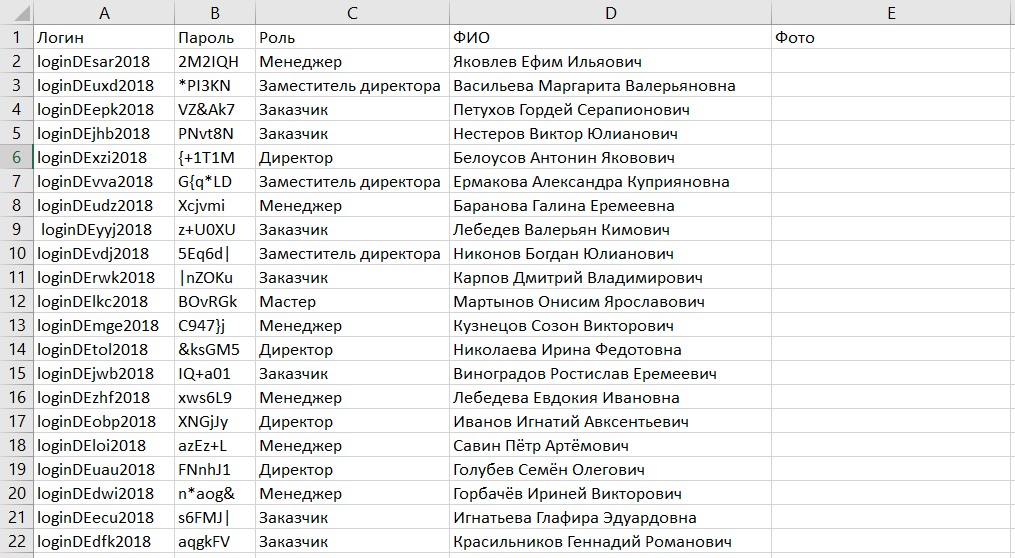


Рисунок 27. Готовая таблица для импорта «Пользователи»

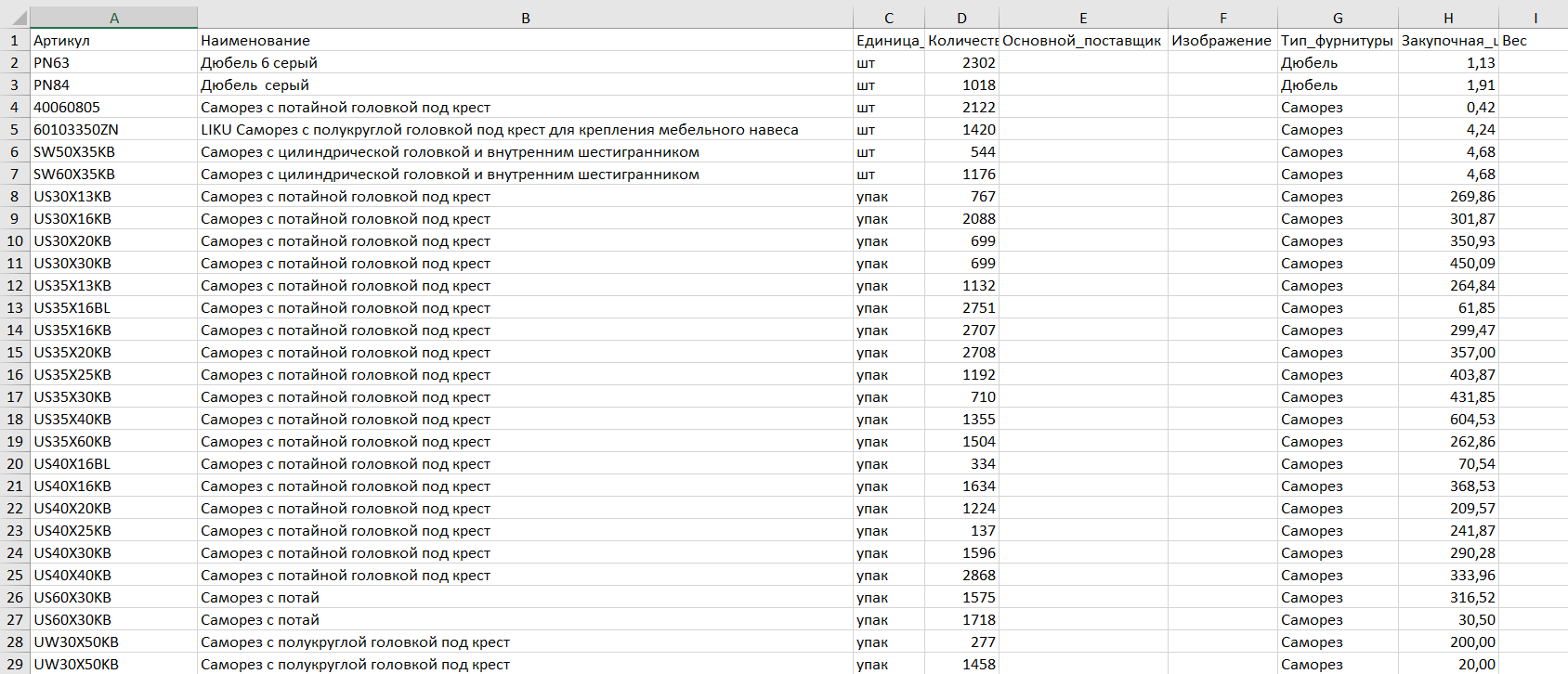


Рисунок 28. Готовая таблица для импорта «Фурнитура»

## 5.2.3 Разработка базы данных «Мебельная фабрика»

USE DE\_Anoshina

CREATE TABLE Поставщики(

Наименование Varchar(50) PRIMARY KEY,

Адрес Varchar(100),

Сроки\_доставки DATE

)

CREATE TABLE Спецификация\_фурнитура(

Изделие Varchar(50) PRIMARY KEY,

Фурнитура Varchar(50),

Количество int

)

CREATE TABLE Спецификация\_материалы(

Изделие Varchar(50) PRIMARY KEY,

Материалы Varchar(50),

Количество int

)

CREATE TABLE Изделие(

Наименование Varchar(50) PRIMARY KEY,

Размеры Varchar(50)

)

CREATE TABLE Спецификация\_сборочная\_единица(

Изделие Varchar(50) PRIMARY KEY,

Сборочная\_единица\_деталь Varchar(50)

)

CREATE TABLE Заказ(

Номер int PRIMARY KEY,

Дата Date,

Наименование\_заказа Varchar(50),

Изделие Varchar(50),

Заказчик Varchar(50),

Ответственный\_менеджер Varchar(50),

Стоимость Varchar(50),

Плановая\_дата\_завершения Date,

Схемы\_чертежи\_заказа Varchar(50)

)

CREATE TABLE Тип\_оборудования(

Наименование Varchar(50),

)

CREATE TABLE Спецификация\_оперции(

Изделие Varchar(50),

Операция Varchar(50),

Порядковый\_номер int,

Тип\_оборудования Varchar(50),

Время\_на\_операцию DateTime

)

CREATE TABLE Оборудование(

Маркировка Varchar(50),

Тип\_оборудования Varchar(50),

Характеристика Varchar(100)

)

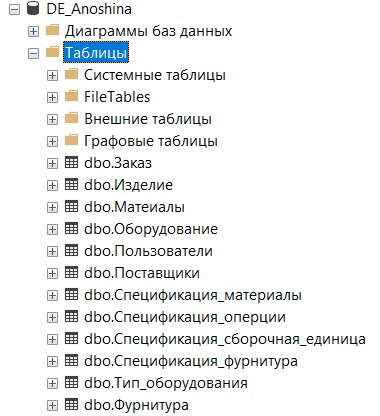


Рисунок 29. Таблицы