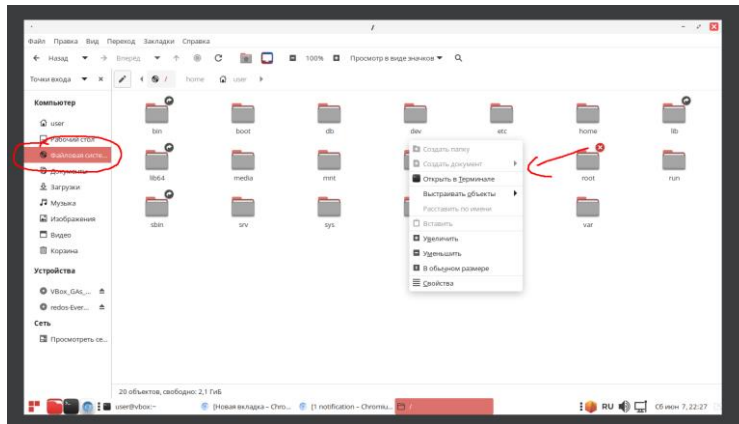


1. Открываем терминал в корневом каталоге

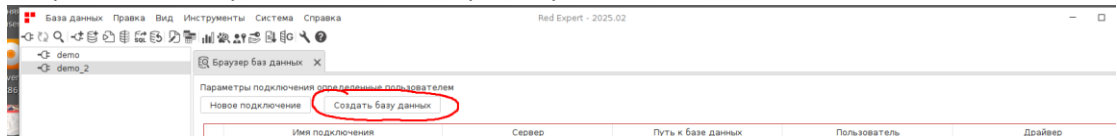


2. Прописываем команды в терминале

```
sudo mkdir /db
```

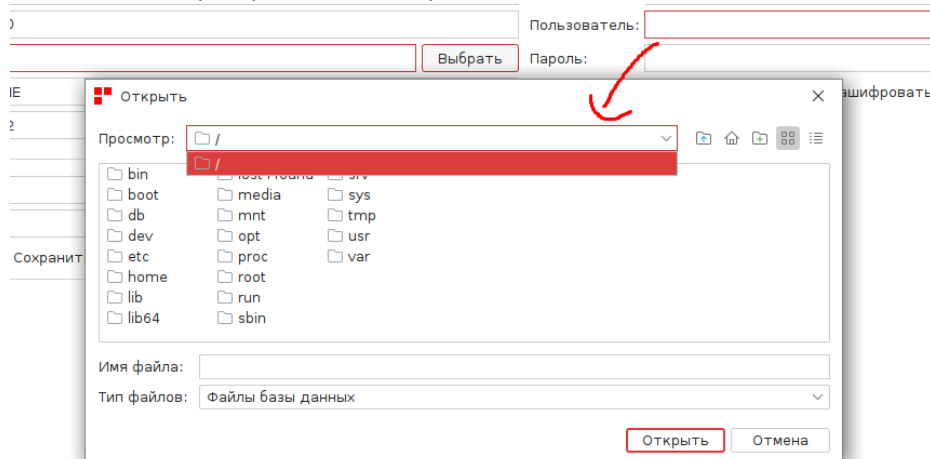
```
sudo sudo chown reddenbase /db
```

3. Открываем RedExpert и создаем новую базу данных

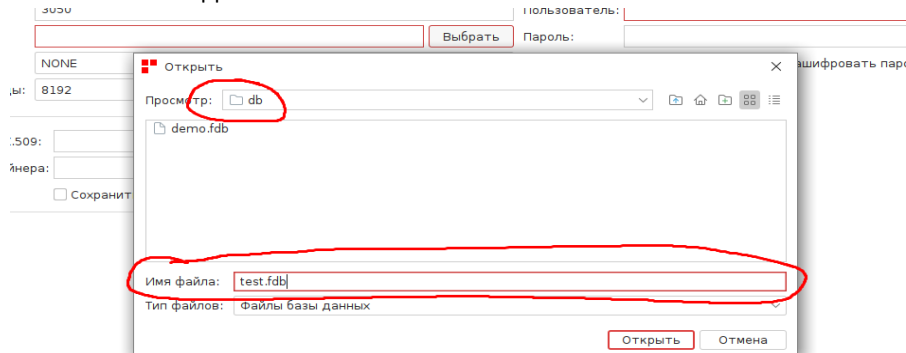


4. Выбираем куда будем сохранять базу данных.

Сначала выбираем корневой каталог, в нём выбираем ранее созданную папку, и пишем название, которое указали во 2 пункте.

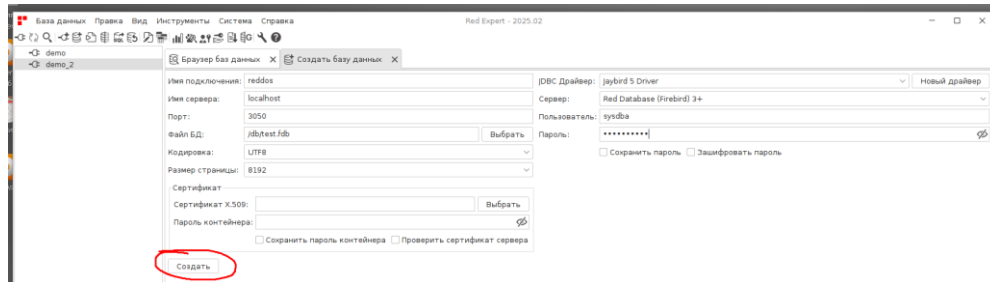


5. Открываем эту папку, и создаем базу данных, в графе «Имя файла», там указываем название базы данных и пишем .fdb



6. Затем пишем:

Имя соединения: какое хотите  
Файл БД: мы выбрали в 5 пункте  
Кодировка: UTF8  
Пользователь: sysdba  
Пароль: student (пароль суперпользователя)



После всех махинаций, у нажимаем кнопку создать и у вас создается БД.

7. У вас слева появилось новое подключение



Тыкаете 2 раза и у вас появляется «редактор запросов»

Туда вписываете вот эту команду:

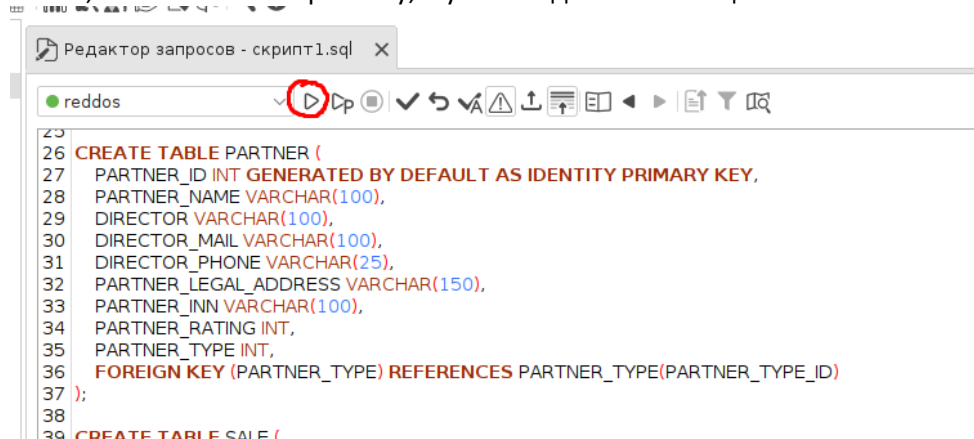
```
CREATE TABLE PRODUCT_TYPE (  
    PRODUCT_TYPE_ID INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
    PRODUCT_TYPE_NAME VARCHAR(100),  
    PRODUCT_TYPE_COEFFICIENT DECIMAL(10,2)  
);  
  
CREATE TABLE MATERIAL_TYPE (  
    MATERIAL_TYPE_ID INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
    MATERIAL_TYPE_NAME VARCHAR(100),  
    PERCENTAGE_DEFECTIVE_MATERIAL DECIMAL(5,4)  
);  
  
CREATE TABLE PARTNER_TYPE (  
    PARTNER_TYPE_ID INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
    PARTNER_TYPE_NAME VARCHAR(100)  
);  
  
CREATE TABLE PRODUCT (  
    PRODUCT_ARTICLE VARCHAR(7) PRIMARY KEY,  
    PRODUCT_NAME VARCHAR(150),  
    MINIMUM_COST_FOR_PARTNER DECIMAL(15,2),  
    PRODUCT_TYPE INT,  
    FOREIGN KEY (PRODUCT_TYPE) REFERENCES PRODUCT_TYPE(PRODUCT_TYPE_ID)  
);  
  
CREATE TABLE PARTNER (  
    PARTNER_ID INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
    PARTNER_NAME VARCHAR(100),  
    DIRECTOR VARCHAR(100),  
    DIRECTOR_MAIL VARCHAR(100),  
    DIRECTOR_PHONE VARCHAR(25),  
    PARTNER_LEGAL_ADDRESS VARCHAR(150),  
    PARTNER_INN VARCHAR(100),  
    PARTNER_RATING INT,
```

```
PARTNER_TYPE INT,
FOREIGN KEY (PARTNER_TYPE) REFERENCES PARTNER_TYPE(PARTNER_TYPE_ID)
);
```

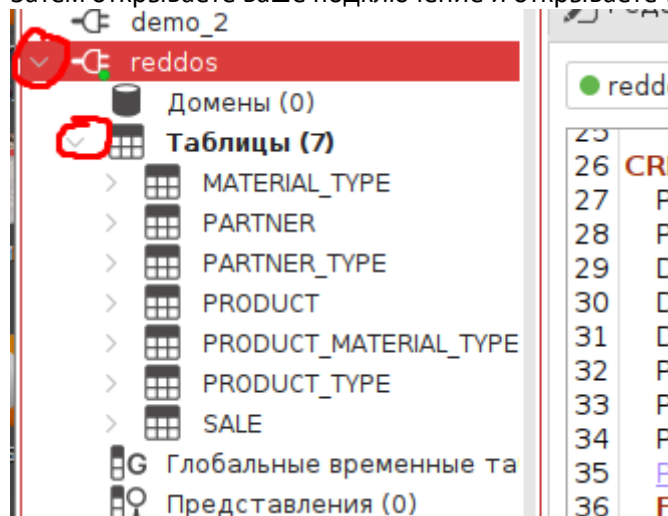
```
CREATE TABLE SALE (
  SALE_ID INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
  PRODUCT_ARTICLE VARCHAR(7),
  PARTNER INT,
  PRODUCT_COUNT INT,
  SALE_DATE DATE,
  FOREIGN KEY (PARTNER) REFERENCES PARTNER(PARTNER_ID),
  FOREIGN KEY (PRODUCT_ARTICLE) REFERENCES PRODUCT(PRODUCT_ARTICLE)
);
```

```
CREATE TABLE PRODUCT_MATERIAL_TYPE (
  PRODUCT_MATERIAL_TYPE_ID INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
  PRODUCT VARCHAR(7),
  MATERIAL_TYPE INT,
  FOREIGN KEY (PRODUCT) REFERENCES PRODUCT(PRODUCT_ARTICLE),
  FOREIGN KEY (MATERIAL_TYPE) REFERENCES MATERIAL_TYPE(MATERIAL_TYPE_ID)
);
```

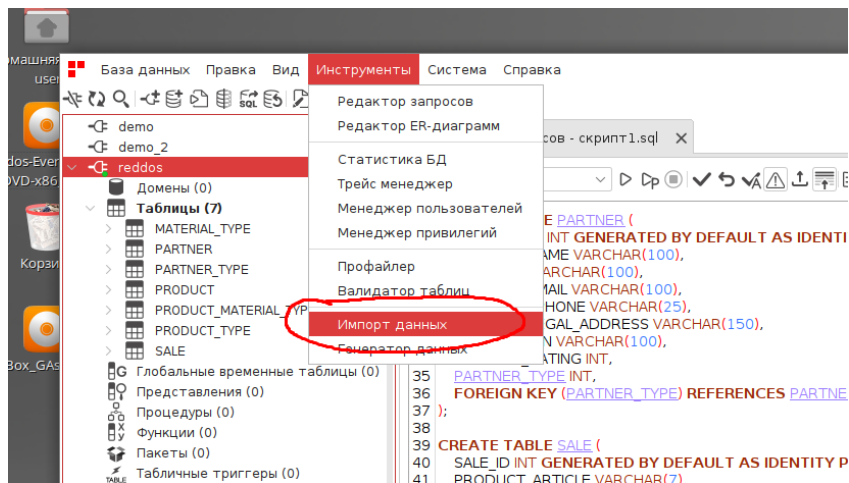
8. После, нажимаете на стрелочку, и у вас создаются таблицы:



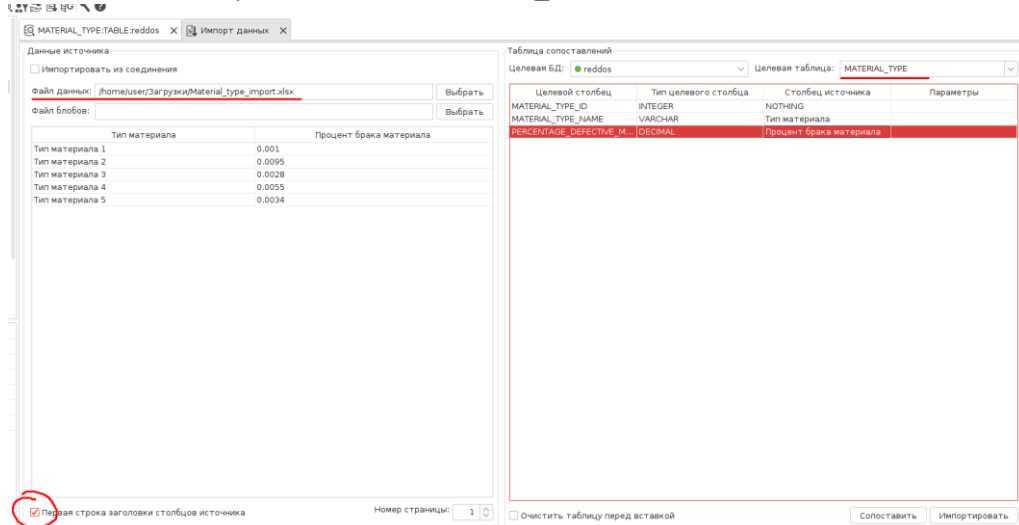
9. Затем открываете ваше подключение и открываете таблицы:



10. Теперь вам надо их все импортировать. Нажимайте «Инструменты» и затем «Импорт данных»

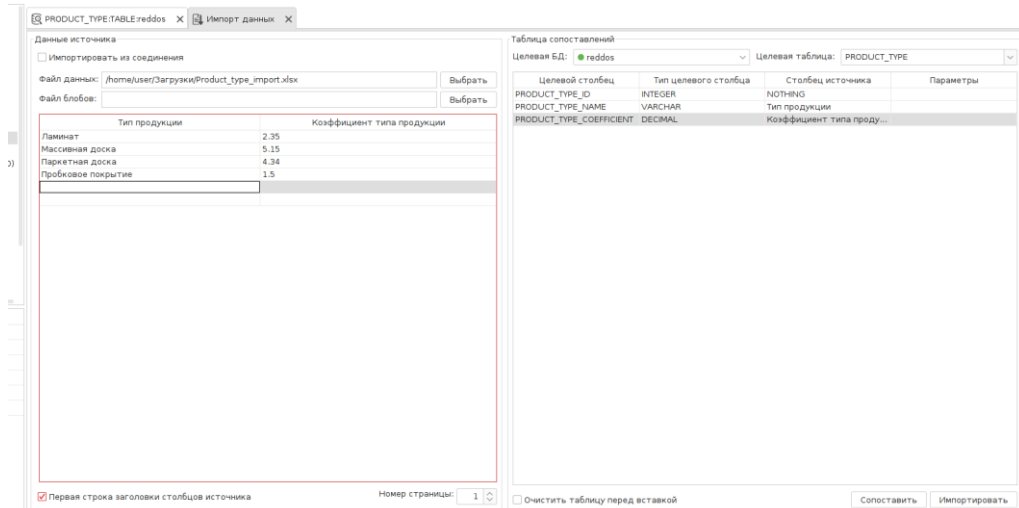


## 11. Начинайте с импорта таблицы «MATERIAL\_TYPE»



Импортируйте

## 12. Импортируем «PRODUCT\_TYPE»



После импорта перейдите в таблицу, и удалите 2 строки пустых. Выделяем, нажимаем на крестик и нажимаем на галочку.

PRODUCT\_TYPE:TABLE:reddos X

Имя таблицы: PRODUCT\_TYPE

Структура Ограничения Индексы Триггеры Привилегии Ссылки Данные SQL Метаданные Зависимости Свойства

Загрузить все

PRODUCT_TYPE_ID	PRODUCT_TYPE_NAME	PRODUCT_TYPE_COEFFICIENT
1	1 Ламинат	2.35
2	2 Массивная доска	5.15
3	3 Паркетная доска	4.34
4	4 Пробковое покрытие	1.50
5	<NULL>	<NULL>
6	<NULL>	<NULL>

13. Импортируем таблицу PRODUCT. В ней есть косяки, заходим в Products.xlsx файл и заменяем. Нужно изменить содержимое файла. «Тип продукции меняем с названия на его id, который был присвоен автоматически при импорте в таблицу PRODUCT\_TYPE. Таблицу артикул меняем с типа int, на строковый.

	А	Б	С	Д	Е
1	Тип продукции	Наименование продукции	Артикул	Минимальная стоимость для партнера	
2		3 Паркетная доска Ясень темный однополосная 14 мм	8758385	4456,90	
3		3 Инженерная доска Дуб Французская елка однополосная 12 мм	8858958	7330,99	
4		1 Ламинат Дуб дымчато-белый 33 класс 12 мм	7750282	1799,33	
5		1 Ламинат Дуб серый 32 класс 8 мм с фаской	7028748	3890,41	
6		4 Пробковое напольное клеевое покрытие 32 класс 4 мм	5012543	5450,59	
7					

Сохраняем и переходим в redexpert

PRODUCT\_TYPE:TABLE:reddos X Импорт данных X

Данные источника

Импортировать из соединения

Файл данных: /home/user/Загрузки/Products\_import.xlsx

Файл блоков:

Тип продукции	Наименование продук...	Артикул	Минимальная стоимость ...
3.0	Паркетная доска Ясен...	8758385	4456.9
3.0	Инженерная доска Ду...	8858958	7330.99
1.0	Ламинат Дуб дымчато...	7750282	1799.33
1.0	Ламинат Дуб серый 32...	7028748	3890.41
4.0	Пробковое напольное ...	5012543	5450.59

Таблица сопоставлений

Целевая БД: reddos

Целевая таблица: PRODUCT

Целевой столбец	Тип целевого столбца	Столбец источника	Параметры
PRODUCT_ARTICLE	VARCHAR	Артикул	
PRODUCT_NAME	VARCHAR	Наименование продукции	
MINIMUM_COST_FOR_PART...	DECIMAL	Минимальная стоимость ...	
PRODUCT_TYPE	INTEGER	Тип продукции	

14. Заходим в таблицу «PARTNER\_TYPE» создаем 4 записи, и заполняем их как на рисунке.

Загрузить все

PARTNER_TYPE_ID	PARTNER_TYPE_NAME
1	1 ЗАО
2	2 ООО
3	3 ПАО
4	4 ОАО

Сохраняем

Открываем partners\_import.xlsx и заменяем как на рисунке

	А	Б	С	
1	Тип партнера	Наименование партнер	Директор	Эле
2		1 База Строитель	Иванова Александра Ивановна	alek
3		2 Паркет 29	Петров Василий Петрович	vpd
4		3 Стройсервис	Соловьев Андрей Николаевич	ans
5		4 Ремонт и отделка	Воробьева Екатерина Валерьевна	ekat
6		1 МонтажПро	Степанов Степан Сергеевич	step
7				

Сохраняем

Переходим в redexpert

Данные источника

☐ Импортировать из соединения

Файл данных: 

Выбрать

Файл блобов: 

Выбрать

Тип пар...	Наимен...	Директор	Электр...	Телефо...	Юридик...	ИИН	Рейтинг
1.0	База Ст...	Иванов...	aleksan...	493 123 ...	652050 ...	2.22245...	7.0
2.0	Паркет ...	Петров ...	vpetrov...	987 123 ...	164500 ...	3.33388...	7.0
3.0	Стройс...	Соловь...	ansolov...	812 223 ...	188910 ...	4.44039...	7.0
4.0	Ремонт ...	Бороб...	ekatem...	444 222 ...	143960 ...	1.11152...	5.0
1.0	Монтаж...	Степан...	stapano...	912 888 ...	209500 ...	5.55243...	10.0

☒ Первая строка заголовки столбцов источника

Номер страницы:

Таблица сопоставлений

Целевая БД:

Целевая таблица:

Целевой столбец	Тип целевого столбца	Столбец источника	Параметры
PARTNER_ID	INTEGER	ИДЕНТИФИКАТОР	
PARTNER_NAME	VARCHAR	Наименование партнера	
DIRECTOR	VARCHAR	Директор	
DIRECTOR_MAIL	VARCHAR	Электронная почта партн...	
DIRECTOR_PHONE	VARCHAR	Телефон партнера	
PARTNER_LEGAL_ADDRESS	VARCHAR	Юридический адрес парт...	
PARTNER_IIN	VARCHAR	ИИН	
PARTNER_RATING	INTEGER	Рейтинг	
PARTNER_TYPE	INTEGER	Тип партнера	

☐ Очистить таблицу перед вставкой

Сопоставить

Импортировать

Импортируем

- В partner\_products\_import.xlsx нужно заменить «продукцию» и «наименование партнера»  
Из product\_import – берём наименование артикулы, и меняем их в нашей таблице в столбце «тип продукции»

Наименование партнёров берём из partner\_import и меняем.

Затем нам нужно поменять дату -> выделяем и выбираем «формат ячеек»

fx

Σ

23.03.2023

A	B	C	D	E	F	G	H
	Наименование партнера	Количество продукции	Дата продажи				
1		15500	23.03.2023				
1		12350	18.12.2023				
1		37400	07.06.2024				
2		35000	02.12.2022				Наименование продукции
2		1250	17.05.2023				Паркетная доска Ясень темный однополосная 14 мм
2		1000	07.06.2024				Инженерная доска Дуб Французская елка однополосная
2		7550	01.07.2024				Ламинат Дуб дымчато-белый 32 класс 12 мм
3		7250	22.04.2024				Ламинат Дуб серый 32 класс 8 мм с фаской
3		2500	05.06.2024				Пробковое напольное клеевое покрытие 32 класс 4 мм
4		59050	20.06.2024				
4		37200	12.06.2024				
4		45000	14.06.2024				
5		50000	19.06.2024				
5		670000	10.07.2024				
5		35000	15.07.2024				
5		25000	12.07.2024				

Вырезать

Ctrl+X

Копировать

Ctrl+C

Вставить

Ctrl+V

Вставить как

Список выбора

Alt+Down

Вставить...

Ctrl++

Удалить...

Ctrl+-

Очистить содержимое...

Backspace

Объединить ячейки

Копировать формат

Очистить форматирование

Ctrl+M

Стили

Вставить комментарий

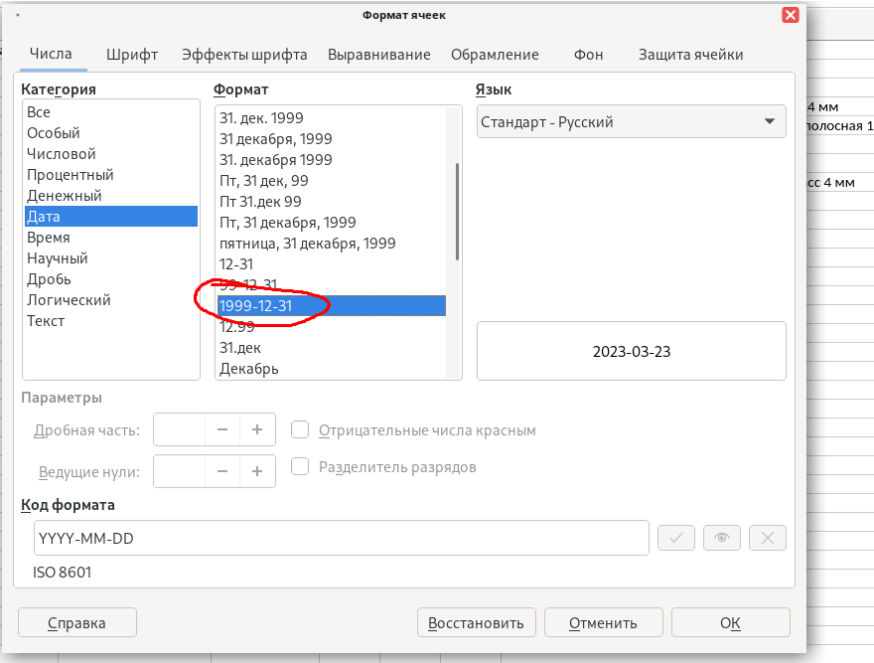
Ctrl+Alt+C

Скрытие

Формат ячеек...

Ctrl+1

Меняем формат ячеек на этот:



Тыкаем ок

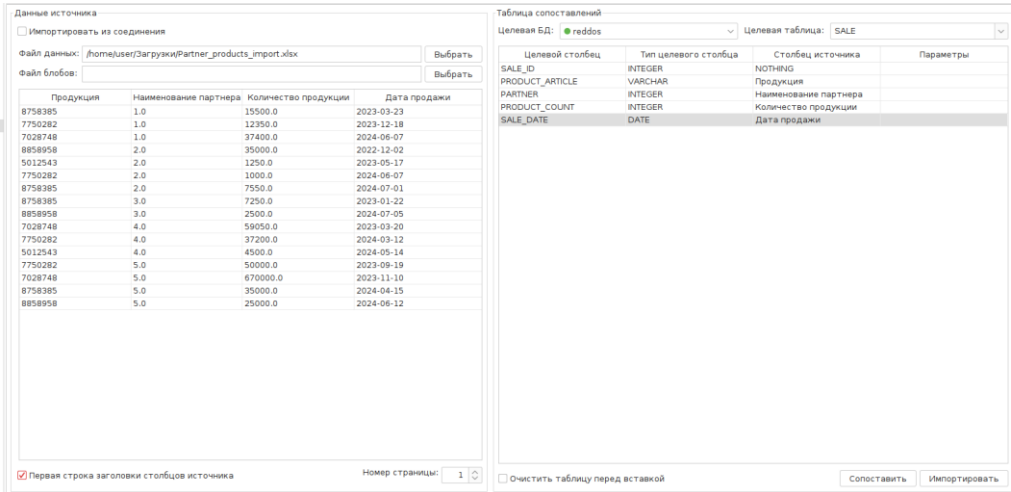
И затем меняем дату на «текст»

Затем выбираем «наименование партнера» и выбираем там тип данных «текст»

По итогу должна быть вот такая таблица:

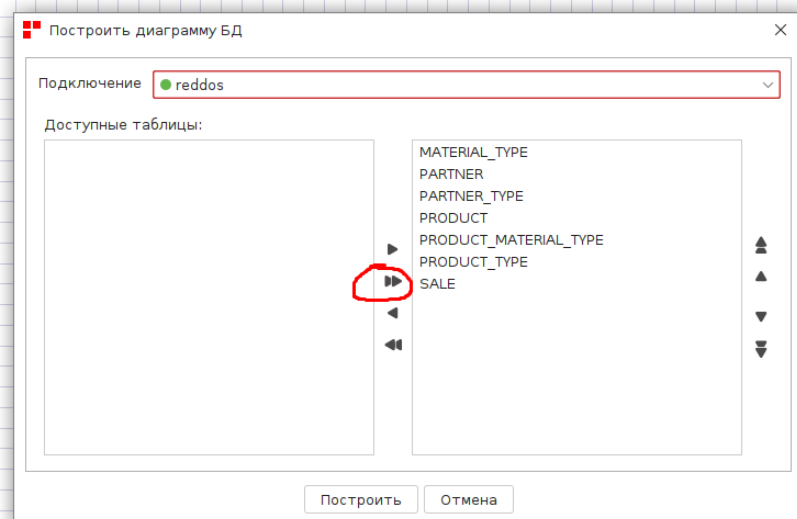
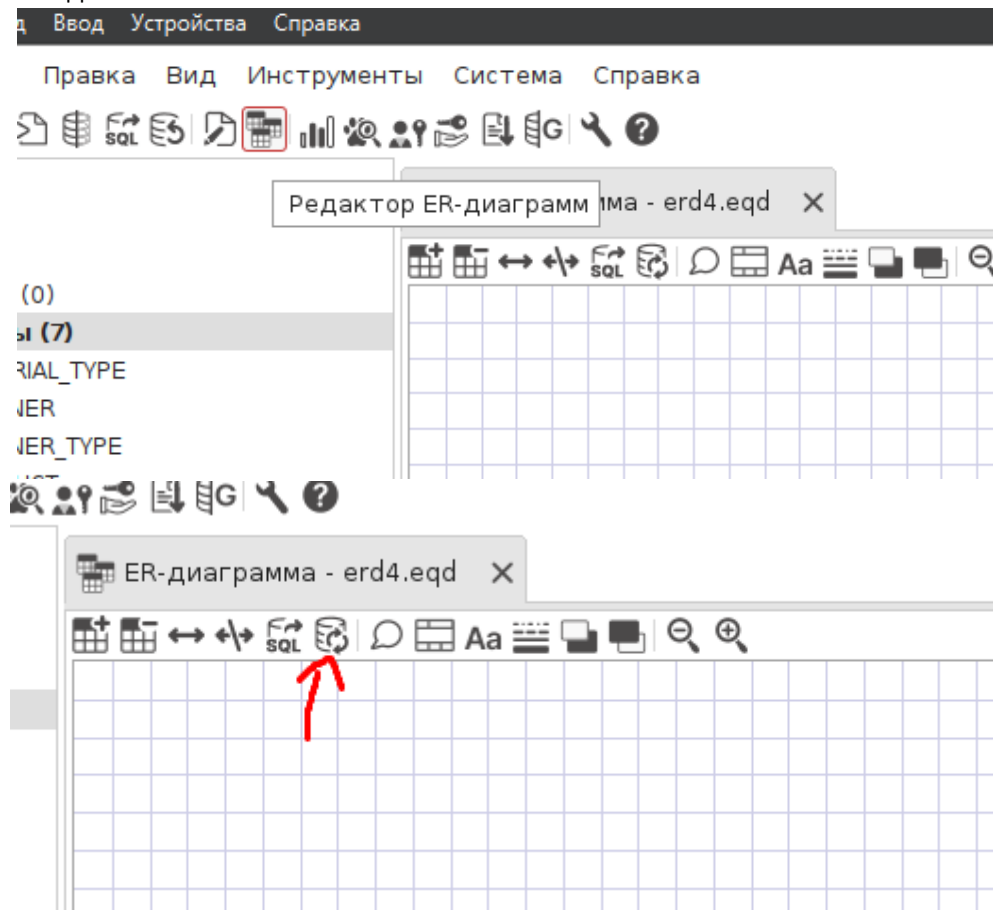
	А	В	С	Д	Е
1	Продукция	Наименование партнера	Количество продукции	Дата продажи	
2	8758385	1	15500	2023-03-23	
3	7750282	1	12350	2023-12-18	
4	7028748	1	37400	2024-06-07	
5	8858958	2	35000	2022-12-02	
6	5012543	2	1250	2023-05-17	
7	7750282	2	1000	2024-06-07	
8	8758385	2	7550	2024-07-01	
9	8758385	3	7250	2023-01-22	
0	8858958	3	2500	2024-07-05	
1	7028748	4	59050	2023-03-20	
2	7750282	4	37200	2024-03-12	
3	5012543	4	4500	2024-05-14	
4	7750282	5	50000	2023-09-19	
5	7028748	5	670000	2023-11-10	
6	8758385	5	35000	2024-04-15	
7	8858958	5	25000	2024-06-12	

Сохраняем и идём в redexpert



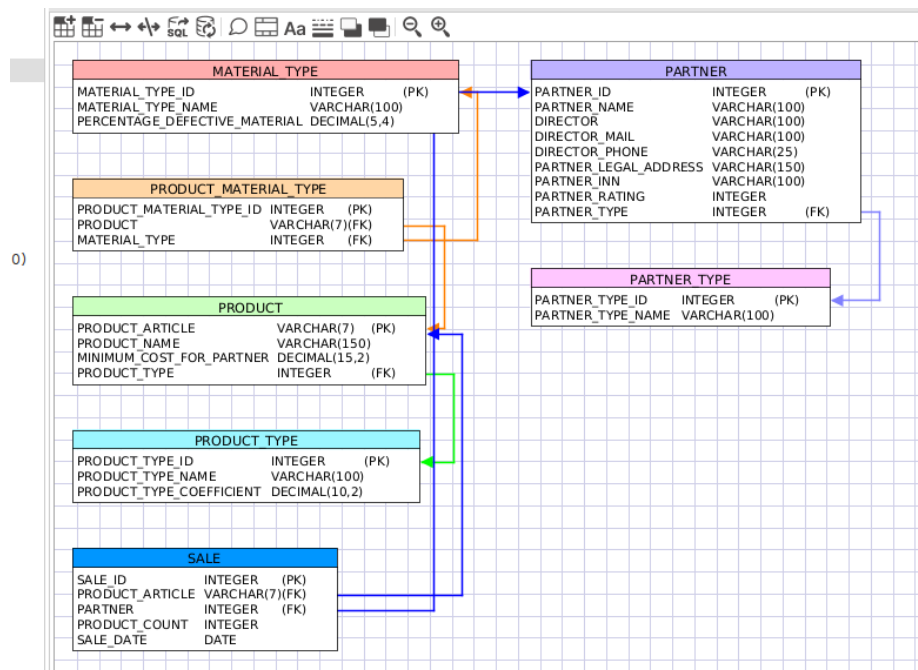
Импортируем

16. Заходим в



Строим





На этом создание базы данных закончено.

## Visual Studio Code

1. Создаем рабочую папку и терминале этой папки прописываем эти команды по очереди

```
dotnet new install Avalonia.Templates
```

```
dotnet new avalonia.mvvm -o DemoApp  
cd DemoApp
```

```
dotnet add package Microsoft.EntityFrameworkCore.Tools --version 9.0.3  
dotnet add package Microsoft.EntityFrameworkCore.Design --version 9.0.3  
dotnet add package FirebirdSql.EntityFrameworkCore.Firebird --version 12.0.0  
dotnet add package Avalonia.ReactiveUI --version 11.2.6  
dotnet add package Avalonia.Xaml.Behaviors --version 11.2.0.14  
dotnet add package MessageBox.Avalonia --version 3.2.0
```

```
dotnet new tool-manifest  
dotnet tool install dotnet-ef --version 9.0.3
```

2. Подключаем базу данных к проекту. Прописываем эти команды. Password пишем student, и меняем название базы данных и корневой каталог.

```
dotnet tool restore
```

```
dotnet tool run dotnet-ef dbcontext scaffold  
"DataSource=localhost;Port=3050;Database=/db/demo.fdb;Username=sysdba;Password=pa  
ssword" FirebirdSql.EntityFrameworkCore.Firebird -o Entities -f
```

3. После успешного подключения, заходим в папку проекта и копируем и создаем поочередно:

Создаем папку Helper -> в ней файл ValidationHelper.cs

**ValidationHelper.cs**

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text.RegularExpressions;  
using System.Threading.Tasks;
```

```
namespace DemoApp.Helpers  
{  
    public class ValidationHelper  
    {  
        /// <summary>  
        /// Проверка ИНН  
        /// </summary>  
        /// <param name="inn">ИНН</param>  
        /// <returns></returns>  
        public static bool IsValidInn(string? inn)  
        {  
            if (string.IsNullOrEmpty(inn)) return false;  
            return Regex.IsMatch(inn, @"^\d{10}$|^\d{12}$");  
        }  
  
        public static bool IsValidEmail(string? email)
```

```

    {
        if (string.IsNullOrWhiteSpace(email)) return false;
        return Regex.IsMatch(email, @"^[^@\s]+@^[^@\s]+\.[^@\s]+$");
    }

    public static bool IsValidPhoneNumber(string? phone)
    {
        if (string.IsNullOrWhiteSpace(phone)) return false;
        return Regex.IsMatch(phone, @"^\+?[78]\s?(?\d{3})?\s?\d{3}[-\s]?d{2}[-\s]?d{2}$");
    }

    public static bool IsValidRating(int? rating)
    {
        {
            return rating is >= 0 and <= 100;
        }
    }
}

```

В Models -> создаем файл Partner.cs

**Partner.cs**

```
using System.Linq;
namespace DemoApp.Models
{
    public class Partner : Entities.Partner
    {
        public Entities.Partner _partner { get; set; }
        public Partner(Entities.Partner partner)
        {
            _partner = partner;
        }
        public string? PartnerTypeDescription =>
            _partner.PartnerTypeNavigation?.PartnerTypeName;
        public string ContactInfo => $" {_partner.DirectorPhone} | {_partner.DirectorMail}";
        public string RatingDisplay => $"Рейтинг: {_partner.PartnerRating ?? 0}";
        public int TotalProductsSold => _partner.Sales?.Sum(s => s.ProductCount) ?? 0;
        public decimal Discount => CalculatePartnerDiscount(TotalProductsSold);
        public string DiscountDisplay => $"Скидка: {Discount:P0}";
        public static decimal CalculatePartnerDiscount(int totalProductsSold)
        {
            return totalProductsSold switch
            {
                < 10000 => 0,
                < 50000 => 0.05m,
                < 300000 => 0.10m,
                _ => 0.15m
            };
        }
    }
}
```

В ViewModels -> в файле MainWindowViewModels.cs -> меняем данные  
**MainWindowViewModel.cs**

```
using Avalonia.Controls;
using CommunityToolkit.Mvvm.ComponentModel;
using DemoApp.Entities;
using DemoApp.Views;

namespace DemoApp.ViewModels;

public partial class MainWindowViewModel : ViewModelBase
{
    private readonly DbDemoFdbContext _context;

    [ObservableProperty]
    private Control? _content;
    public void ShowContent(Control content)
    {
        Content = content;
    }
    public void ShowPartnersList()
    {
        Content = new PartnersView
        {
            DataContext = new PartnersViewModel(_context, this)
        };
    }

    public MainWindowViewModel()
    {
        _context = new DbDemoFdbContext();
        ShowPartnersList();
    }

    public void ShowEditPartner(EditPartnerView editControl)
    {
        Content = editControl;
    }
}
```

В ViewModels создаем файл -> EditPartnerViewModel.cs

**EditPartnerViewModel.cs**

```
using System;
using System.Collections.Generic;
using System.Collections.ObjectModel;
using System.Linq;
using System.Threading.Tasks;
using CommunityToolkit.Mvvm.ComponentModel;
using CommunityToolkit.Mvvm.Input;
using DemoApp.Entities;
using DemoApp.Helpers;
using MsBox.Avalonia;
using MsBox.Avalonia.Enums;

namespace DemoApp.ViewModels
{
    public partial class EditPartnerViewModel : ViewModelBase
    {
        private readonly DbDemoFdbContext _context;
        private readonly PartnersViewModel? _parentViewModel;
        private readonly Partner _partnerEntity;
        [ObservableProperty]
        private string? _partnerName;

        [ObservableProperty]
        private string? _partnerInn;

        [ObservableProperty]
        private string? _director;

        [ObservableProperty]
        private string? _directorPhone;

        [ObservableProperty]
        private string? _directorMail;

        [ObservableProperty]
        private int? _partnerRating;

        [ObservableProperty]
        private int? _selectedPartnerType;

        [ObservableProperty]
        private ObservableCollection<PartnerType> _partnerTypes = new();

        [ObservableProperty]
        private string? _errorMessage;
        public EditPartnerViewModel(DbDemoFdbContext context, Partner? partner = null,
PartnersViewModel? parentViewModel = null)
        {
            _context = context;
            _parentViewModel = parentViewModel;
            _partnerEntity = partner ?? new Partner();
        }
    }
}
```

```

        LoadPartnerTypes();
        LoadPartnerData();
    }

    private void LoadPartnerTypes()
    {
        var types = _context.PartnerTypes.ToList();
        PartnerTypes = new ObservableCollection<PartnerType>(types);
    }
    private void LoadPartnerData()
    {
        PartnerName = _partnerEntity.PartnerName;
        PartnerInn = _partnerEntity.PartnerInn;
        Director = _partnerEntity.Director;
        DirectorPhone = _partnerEntity.DirectorPhone;
        DirectorMail = _partnerEntity.DirectorMail;
        PartnerRating = _partnerEntity.PartnerRating;
        SelectedPartnerType = _partnerEntity.PartnerType;
    }

    private bool ValidateData()
    {
        if (string.IsNullOrEmpty(PartnerName))
        {
            ErrorMessage = "Название партнера обязательно для заполнения";
            return false;
        }

        if (!ValidationHelper.IsValidInn(PartnerInn))
        {
            ErrorMessage = "ИНН должен содержать 10 или 12 цифр";
            return false;
        }

        if (string.IsNullOrEmpty(Director))
        {
            ErrorMessage = "Имя директора обязательно для заполнения";
            return false;
        }

        if (!ValidationHelper.IsValidPhoneNumber(DirectorPhone))
        {
            ErrorMessage = "Неверный формат номера телефона. Пример: +7(999)999-99-99";
            return false;
        }

        if (!ValidationHelper.IsValidEmail(DirectorMail))
        {
            ErrorMessage = "Неверный формат электронной почты";
            return false;
        }
    }

```

```

if (!ValidationHelper.IsValidRating(PartnerRating))
{
    ErrorMessage = "Рейтинг должен быть от 0 до 100";
    return false;
}

if (SelectedPartnerType == null)
{
    ErrorMessage = "Выберите тип партнера";
    return false;
}

ErrorMessage = null;
return true;
}

```

```

[RelayCommand]
private async Task SaveAsync()
{
    if (!ValidateData())
    {
        var box = MessageBoxManager
        .GetMessageBoxStandard("Error", ErrorMessage,
        ButtonEnum.Ok);

        var result = await box.ShowAsync();
        return;
    }
}

```

```

_partnerEntity.PartnerName = PartnerName;
_partnerEntity.PartnerInn = PartnerInn;
_partnerEntity.Director = Director;
_partnerEntity.DirectorPhone = DirectorPhone;
_partnerEntity.DirectorMail = DirectorMail;
_partnerEntity.PartnerRating = PartnerRating;
_partnerEntity.PartnerType = SelectedPartnerType;

if (_partnerEntity.PartnerId == 0)
{
    _context.Partners.Add(_partnerEntity);
}

_context.SaveChanges();
_parentViewModel?.LoadPartners();
Cancel();
}

```

```

[RelayCommand]
private void Cancel()
{
    _parentViewModel?._mainViewModel.ShowPartnersList();
}

```



}

}

}

В ViewModels создаем файл -> PartnersViewModel.cs

**PartnersViewModel.cs**

```
using System.Collections.ObjectModel;
using System.Linq;
using CommunityToolkit.Mvvm.ComponentModel;
using CommunityToolkit.Mvvm.Input;
using DemoApp.Entities;
using DemoApp.Views;
using Microsoft.EntityFrameworkCore;

namespace DemoApp.ViewModels
{
    public partial class PartnersViewModel : ViewModelBase
    {
        [ObservableProperty]
        private ObservableCollection<Models.Partner> _partners = new();

        [ObservableProperty]
        private Models.Partner? _selectedPartner;
        private readonly DbDemoFdbContext _context;
        internal readonly MainWindowViewModel _mainViewModel;

        public PartnersViewModel(DbDemoFdbContext context, MainWindowViewModel
mainViewModel)
        {
            _context = context;
            _mainViewModel = mainViewModel;
            LoadPartners();
        }

        public void LoadPartners()
        {
            var entities = _context.Partners.Include(p => p.Sales).Include(p =>
p.PartnerTypeNavigation).ToList();
            Partners = new ObservableCollection<Models.Partner>(
                entities.Select(e => new Models.Partner(e))
            );
        }

        [RelayCommand]
        private void EditPartner()
        {
            if (SelectedPartner == null) return;
            var partnerEntity = _context.Partners.Find(SelectedPartner._partner.PartnerId);
            if (partnerEntity == null) return;

            var viewModel = new EditPartnerViewModel(_context, partnerEntity, this);
            var editControl = new EditPartnerView { DataContext = viewModel };
        }
    }
}
```

```
        _mainViewModel.ShowEditPartner(editControl);  
    }  
}
```

```
[RelayCommand]  
private void AddPartner()  
{  
    var viewModel = new EditPartnerViewModel(_context, parentViewModel: this);  
    var editControl = new EditPartnerView { DataContext = viewModel };  
    _mainViewModel.ShowEditPartner(editControl);  
}
```

```
[RelayCommand]  
private void ViewSalesHistory(Partner? partner)  
{  
    //история продаж  
}  
}  
}
```

## B Views EditPartnerView.axaml

```
<UserControl xmlns="https://github.com/avaloniaui"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d" d:DesignWidth="800" d:DesignHeight="450"
    x:Class="DemoApp.Views.EditPartnerView"
    xmlns:vm="using:DemoApp.ViewModels"
    x:DataType="vm:EditPartnerViewModel">
    <Grid Margin="20">
        <StackPanel Spacing="10">
            <TextBlock Text="Редактирование партнера" FontSize="24" FontWeight="Bold" />
            <TextBlock Text="{Binding ErrorMessage}"
                Foreground="Red"
                IsVisible="{Binding ErrorMessage, Converter={x:Static ObjectConverters.IsNotNull}}"
                TextWrapping="Wrap"/>
        </StackPanel>
        <Grid ColumnDefinitions="150,*"
            RowDefinitions="Auto,Auto,Auto,Auto,Auto,Auto,Auto,Auto,Auto,Auto">
            <TextBlock Grid.Row="0" Grid.Column="0" Text="Название:" />
            <TextBox Grid.Row="0" Grid.Column="1" Text="{Binding PartnerName}" />
            <TextBlock Grid.Row="1" Grid.Column="0" Text="ИНН:" />
            <TextBox Grid.Row="1" Grid.Column="1" Text="{Binding PartnerInn}" />
            <TextBlock Grid.Row="2" Grid.Column="0" Text="Директор:" />
            <TextBox Grid.Row="2" Grid.Column="1" Text="{Binding Director}" />
            <TextBlock Grid.Row="3" Grid.Column="0" Text="Телефон:" />
            <MaskedTextBox Grid.Row="3" Grid.Column="1" Text="{Binding DirectorPhone}"
                Mask="+7 (000) 000 0000" />
            <TextBlock Grid.Row="4" Grid.Column="0" Text="Email:" />
            <TextBox Grid.Row="4" Grid.Column="1" Text="{Binding DirectorMail}"
                Watermark="example@domain.com" />
            <TextBlock Grid.Row="5" Grid.Column="0" Text="Рейтинг:" />
            <NumericUpDown Grid.Row="5" Grid.Column="1" Value="{Binding PartnerRating}"
                Minimum="0" Maximum="100" />
            <TextBlock Grid.Row="6" Grid.Column="0" Text="Тип партнера:" />
            <ComboBox Grid.Row="6" Grid.Column="1"
                ItemsSource="{Binding PartnerTypes}"
                SelectedValue="{Binding SelectedPartnerType}"
                SelectedValueBinding="{Binding PartnerTypeId}"
                DisplayMemberBinding="{Binding PartnerTypeName}" />
        </Grid>
        <StackPanel Orientation="Horizontal" Spacing="10">
            <Button Content="Сохранить" Command="{Binding SaveCommand}" />
            <Button Content="Отмена" Command="{Binding CancelCommand}" />
        </StackPanel>
    </Grid>
</UserControl>
```

```

</UserControl>
B Views EditPartnerView.axaml.cs
using Avalonia;
using Avalonia.Controls;
using Avalonia.Markup.Xaml;

namespace DemoApp.Views;

public partial class EditPartnerView : UserControl
{
    public EditPartnerView()
    {
        InitializeComponent();
    }
}

```

#### B Views PartnersView.axaml

```

<UserControl xmlns="https://github.com/avaloniaui"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d" d:DesignWidth="800" d:DesignHeight="450"
    x:Class="DemoApp.Views.PartnersView"
    xmlns:vm="using:DemoApp.ViewModels"
    x:DataType="vm:PartnersViewModel">
    <Grid Margin="20">
        <StackPanel Spacing="10">
            <TextBlock Text="Список партнеров" FontSize="24" FontWeight="Bold" />
            <StackPanel Orientation="Horizontal" Spacing="10">
                <Button Content="Добавить партнера" Command="{Binding AddPartnerCommand}" />
                <Button Content="Редактировать" Command="{Binding EditPartnerCommand}" />
            </StackPanel>
            <ListBox ItemsSource="{Binding Partners}" SelectedItem="{Binding SelectedPartner}"
                MaxHeight="800">
                <Interaction.Behaviors>
                    <EventTriggerBehavior EventName="SelectionChanged">
                        <InvokeCommandAction Command="{Binding EditPartnerCommand}"
                            CommandParameter="{Binding SelectedPartner}" />
                    </EventTriggerBehavior>
                </Interaction.Behaviors>
                <ListBox.ItemTemplate>
                    <DataTemplate>
                        <StackPanel Orientation="Horizontal" Margin="5" HorizontalAlignment="Left">
                            <StackPanel Orientation="Vertical" Margin="5" HorizontalAlignment="Left">
                                <StackPanel Orientation="Horizontal" HorizontalAlignment="Left">
                                    <TextBlock Text="{Binding PartnerTypeDescription}" TextWrapping="Wrap" />
                                    <TextBlock Text=" | " TextWrapping="Wrap" />
                                    <TextBlock Text="{Binding _partner.PartnerName}" TextWrapping="Wrap"
                                        Width="250" />
                                </StackPanel>
                            </StackPanel>
                        </StackPanel>
                    </DataTemplate>
                </ListBox.ItemTemplate>
            </ListBox>
        </StackPanel>
    </Grid>

```

```

        </StackPanel>
        <TextBlock Text="{Binding _partner.Director}" TextWrapping="Wrap" Width="250"
HorizontalAlignment="Left" />
        <TextBlock Text="{Binding _partner.DirectorPhone}" TextWrapping="Wrap"
Width="250" HorizontalAlignment="Left" />
        <TextBlock Text="{Binding RatingDisplay}" TextWrapping="Wrap" Width="250"
HorizontalAlignment="Left" />
        </StackPanel>
        <StackPanel Orientation="Vertical" Spacing="10">
            <TextBlock Text="{Binding DiscountDisplay}" TextWrapping="Wrap" Width="150" />
            <Button Content="История продаж" />
        </StackPanel>
    </StackPanel>
</DataTemplate>
</ListBox.ItemTemplate>
</ListBox>

</StackPanel>
</Grid>

</UserControl>

```

**B Views PartnerView.axaml.cs**

```

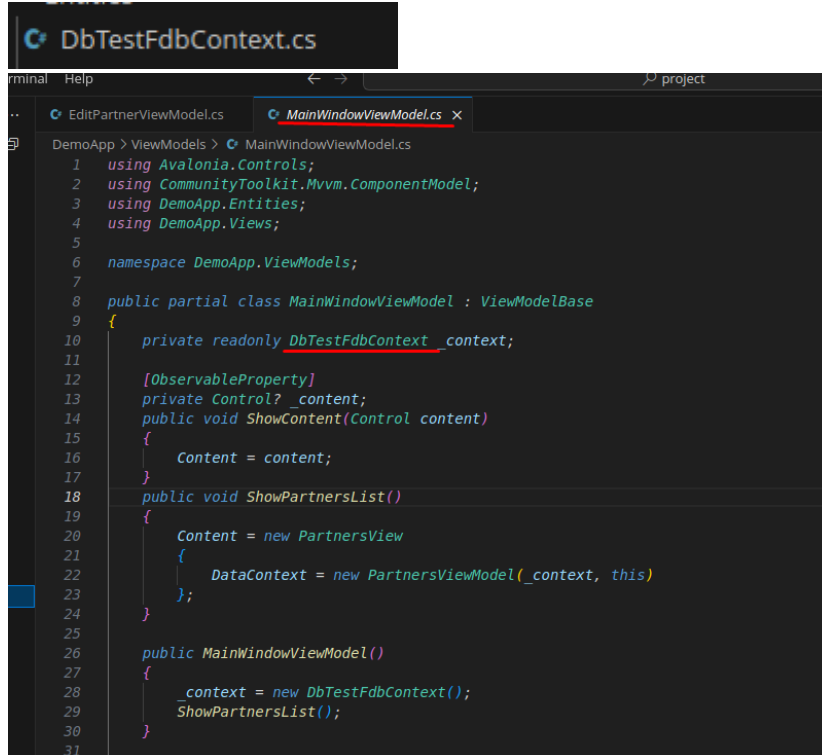
using Avalonia;
using Avalonia.Controls;
using Avalonia.Markup.Xaml;

namespace DemoApp.Views;

public partial class PartnersView : UserControl
{
    public PartnersView()
    {
        InitializeComponent();
    }
}

```

Во всех файлах где подчеркивается красным, меняем где подчёркивается красным на название вашего файла бд.



```
DbTestFdbContext.cs

terminal Help project
EditPartnerViewModel.cs MainWindowViewModel.cs x
DemoApp > ViewModels > MainWindowViewModel.cs
1 using Avalonia.Controls;
2 using CommunityToolkit.Mvvm.ComponentModel;
3 using DemoApp.Entities;
4 using DemoApp.Views;
5
6 namespace DemoApp.ViewModels;
7
8 public partial class MainWindowViewModel : ViewModelBase
9 {
10     private readonly DbTestFdbContext _context;
11
12     [ObservableProperty]
13     private Control? _content;
14     public void ShowContent(Control content)
15     {
16         Content = content;
17     }
18     public void ShowPartnersList()
19     {
20         Content = new PartnersView
21         {
22             DataContext = new PartnersViewModel(_context, this)
23         };
24     }
25
26     public MainWindowViewModel()
27     {
28         _context = new DbTestFdbContext();
29         ShowPartnersList();
30     }
31 }
```