Create database Shadr

CREATE TABLE Administrators (

Id INT PRIMARY KEY IDENTITY(1,1),

FullName VARCHAR(max),

Email VARCHAR(max)

);

CREATE TABLE Designers (

Id INT PRIMARY KEY IDENTITY(1,1),

[Type] VARCHAR(MAX),

[Name] VARCHAR(MAX),

[Description] VARCHAR(MAX),

Email VARCHAR (MAX),

StandardNumber VARCHAR(MAX)

);

CREATE TABLE Programmists (

Id INT PRIMARY KEY IDENTITY(1,1),

PartnerTypeId VARCHAR (MAX),

CompanyName VARCHAR(MAX),

LegalAddress VARCHAR(MAX),

INN VARCHAR(MAX),

Phone VARCHAR(MAX),

Email VARCHAR(MAX)

);

CREATE TABLE Exporters (

Id INT PRIMARY KEY IDENTITY(1,1),

[Name] VARCHAR(MAX)

);

INSERT INTO Exporters ([Name]) VALUES

('git init

git add -A

git commit -m «Merge”

git push(настроить глобал)(если ошибки возникли,то можно удалить все и запушить,либо git push —force)

git pull');

INSERT INTO Administrators (FullName, Email) VALUES

('CREATE DATABASE Mebel;

USE Mebel;

CREATE TABLE PartnerTypes (

Id INT PRIMARY KEY IDENTITY(1,1),

[Name] VARCHAR(100) NOT NULL

);

CREATE TABLE Partners (

Id INT PRIMARY KEY IDENTITY(1,1),

PartnerTypeId INT NOT NULL,

CompanyName VARCHAR(255) NOT NULL,

LegalAddress VARCHAR(255),

INN VARCHAR(12),

DirectorFullName VARCHAR(255),

Phone VARCHAR(20),

Email VARCHAR(100),

Rating INT,

FOREIGN KEY (PartnerTypeId) REFERENCES PartnerTypes(Id)

);

CREATE TABLE Managers (

Id INT PRIMARY KEY IDENTITY(1,1),

FullName VARCHAR(255) NOT NULL,

Email VARCHAR(100),

Phone VARCHAR(20)

);

CREATE TABLE Orders (

Id INT PRIMARY KEY IDENTITY(1,1),

PartnerId INT NOT NULL,

ManagerId INT NOT NULL,

[Status] VARCHAR(50),

CreationDate DATE,

PaymentDate DATE,

DeliveryDate DATE,

TotalAmount INT,

FOREIGN KEY (PartnerId) REFERENCES Partners(Id),

FOREIGN KEY (ManagerId) REFERENCES Managers(Id)

);

CREATE TABLE Suppliers (

Id INT PRIMARY KEY IDENTITY(1,1),

[Name] VARCHAR(255),

INN VARCHAR(12)

);

CREATE TABLE MaterialTypes (

Id INT PRIMARY KEY IDENTITY(1,1),

[Name] VARCHAR(100) NOT NULL

);

CREATE TABLE Materials (

Id INT PRIMARY KEY IDENTITY(1,1),

[Name] VARCHAR(100),

MaterialTypeId INT NOT NULL,

SupplierId INT NOT NULL,

Unit VARCHAR(20),

QuantityPerPack INT,

[Description] VARCHAR(1000),

FOREIGN KEY (SupplierId) REFERENCES Suppliers(Id),

FOREIGN KEY (MaterialTypeId) REFERENCES MaterialTypes(Id)

);

CREATE TABLE Products (

Id INT PRIMARY KEY IDENTITY(1,1),

[Type] VARCHAR(50),

[Name] VARCHAR(100),

[Description] VARCHAR(1000),

MinPartnerPrice INT,

PackageLength INT,

PackageWidth INT,

PackageHeight INT,

WeightWithoutPackage INT,

WeightWithPackage INT,

StandardNumber VARCHAR(50),

CostPrice INT

);

CREATE TABLE OrderItems (

Id INT PRIMARY KEY IDENTITY(1,1),

OrderId INT NOT NULL,

ProductId INT NOT NULL,

Quantity INT,

UnitPrice INT,

ProductionDate DATE,

FOREIGN KEY (OrderId) REFERENCES Orders(Id),

FOREIGN KEY (ProductId) REFERENCES Products(Id)

);

CREATE TABLE Roles (

Id INT PRIMARY KEY IDENTITY(1,1),

[Name] VARCHAR(50) NOT NULL

);

CREATE TABLE Users (

Id INT PRIMARY KEY IDENTITY(1,1),

Username VARCHAR(50) NOT NULL,

[Password] VARCHAR(100) NOT NULL,

RoleId INT NOT NULL,

FOREIGN KEY (RoleId) REFERENCES Roles(Id)

);','

INSERT INTO PartnerTypes ([Name]) VALUES

("Оптовый покупатель"),

("Розничный покупатель");

INSERT INTO MaterialTypes ([Name]) VALUES

("Древесина"),

("Металл"),

("Пластик"),

("Ткань");

INSERT INTO Suppliers ([Name], INN) VALUES

("ООО СтройМатериалы", "123456789012"),

("ЗАО МеталлСнаб", "987654321098");

INSERT INTO Materials ([Name], MaterialTypeId, SupplierId, Unit, QuantityPerPack, [Description]) VALUES

("Доска обрезная", 1, 1, "шт", 50, "Высококачественная обрезная доска из сосны"),

("Профиль алюминиевый", 2, 2, "метр", 100, "Алюминиевый профиль для каркасов"),

("Пластиковая панель", 3, 1, "лист", 10, "Легкие пластиковые панели для отделки");

INSERT INTO Products ([Type], [Name], [Description], MinPartnerPrice, PackageLength, PackageWidth, PackageHeight, WeightWithoutPackage, WeightWithPackage, StandardNumber, CostPrice) VALUES

("Стул", "Стул деревянный", "Удобный деревянный стул для кухни", 1500, 40, 40, 90, 5000, 5200, "СТД-001", 1000),

("Стол", "Стол обеденный", "Обеденный стол из массива древесины", 5500, 150, 90, 75, 20000, 20500, "СТД-002", 4000);

INSERT INTO Roles ([Name]) VALUES ("Admin"), ("Manager"), ("Employee");

INSERT INTO Users (Username, Password, RoleId) VALUES

("admin", "admin", 1),

("manager", "man", 2),

("employee", "empl", 3);');

INSERT INTO Designers ( [Type], [Name], [Description], Email, StandardNumber) VALUES

(' <Grid>

<Label VerticalAlignment="Center" Content="Логин" HorizontalAlignment="Center" Margin="0,0,0,100"></Label>

<Label VerticalAlignment="Center" Content="Пароль" HorizontalAlignment="Center" Margin="0,0,0,0"></Label>

<TextBox x:Name="LogTB" Width="120" Height="30" VerticalAlignment="Center" HorizontalAlignment="Center" Margin="0,0,0,50"></TextBox>

<PasswordBox x:Name="PaswwordTB" Width="120" Height="30" VerticalAlignment="Center" HorizontalAlignment="Center" Margin="0,0,0,-50"></PasswordBox>

<Button x:Name="LoginButton" Width="120" Height="30" Content="Войти" VerticalAlignment="Center" HorizontalAlignment="Center" Margin="0,0,0,-150" Click="LoginButton\_Click" ></Button>

</Grid>', '<Grid>

<DataGrid x:Name="ProductGrid" Width="700" Height="300"

HorizontalAlignment="Center" VerticalAlignment="Center"

AutoGenerateColumns="True" IsReadOnly="True"/>

<Button Content="Добавить" Width="100" Height="30"

HorizontalAlignment="Left" VerticalAlignment="Bottom"

Margin="50,0,0,30" Click="AddButton\_Click"/>

<Button Content="Редактировать" Width="100" Height="30"

HorizontalAlignment="Right" VerticalAlignment="Bottom"

Margin="0,0,50,30" Click="EditButton\_Click"/>

</Grid>', '<Grid>

<DataGrid x:Name="ProductGrid" Width="700" Height="300"

HorizontalAlignment="Center" VerticalAlignment="Center"

AutoGenerateColumns="True" IsReadOnly="True"/>

<Button Content="Обновить" Width="100" Height="30"

HorizontalAlignment="Left" VerticalAlignment="Bottom"

Margin="50,0,0,30" Click="RefreshButton\_Click"/>

<Button Content="Выйти" Width="100" Height="30"

HorizontalAlignment="Right" VerticalAlignment="Bottom"

Margin="0,0,50,30" Click="LogoutButton\_Click"/>

</Grid>', '<Grid Margin="20">

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="150"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Label Content="Название:" Grid.Row="0" Grid.Column="0" VerticalAlignment="Center"/>

<TextBox x:Name="NameTextBox" Grid.Row="0" Grid.Column="1" Margin="0,5"/>

<Label Content="Тип:" Grid.Row="1" Grid.Column="0" VerticalAlignment="Center"/>

<TextBox x:Name="TypeTextBox" Grid.Row="1" Grid.Column="1" Margin="0,5"/>

<Label Content="Описание:" Grid.Row="2" Grid.Column="0" VerticalAlignment="Center"/>

<TextBox x:Name="DescriptionTextBox" Grid.Row="2" Grid.Column="1" Margin="0,5"/>

<Label Content="Мин. цена:" Grid.Row="3" Grid.Column="0" VerticalAlignment="Center"/>

<TextBox x:Name="MinPartnerPriceTextBox" Grid.Row="3" Grid.Column="1" Margin="0,5"/>

<Label Content="Длина упаковки:" Grid.Row="4" Grid.Column="0" VerticalAlignment="Center"/>

<TextBox x:Name="PackageLengthTextBox" Grid.Row="4" Grid.Column="1" Margin="0,5"/>

<Label Content="Ширина упаковки:" Grid.Row="5" Grid.Column="0" VerticalAlignment="Center"/>

<TextBox x:Name="PackageWidthTextBox" Grid.Row="5" Grid.Column="1" Margin="0,5"/>

<Label Content="Высота упаковки:" Grid.Row="6" Grid.Column="0" VerticalAlignment="Center"/>

<TextBox x:Name="PackageHeightTextBox" Grid.Row="6" Grid.Column="1" Margin="0,5"/>

<Label Content="Вес без упаковки:" Grid.Row="7" Grid.Column="0" VerticalAlignment="Center"/>

<TextBox x:Name="WeightWithoutPackageTextBox" Grid.Row="7" Grid.Column="1" Margin="0,5"/>

<Label Content="Вес с упаковкой:" Grid.Row="8" Grid.Column="0" VerticalAlignment="Center"/>

<TextBox x:Name="WeightWithPackageTextBox" Grid.Row="8" Grid.Column="1" Margin="0,5"/>

<Label Content="Номер стандарта:" Grid.Row="9" Grid.Column="0" VerticalAlignment="Center"/>

<TextBox x:Name="StandardNumberTextBox" Grid.Row="9" Grid.Column="1" Margin="0,5"/>

<Label Content="Себестоимость:" Grid.Row="10" Grid.Column="0" VerticalAlignment="Center"/>

<TextBox x:Name="CostPriceTextBox" Grid.Row="10" Grid.Column="1" Margin="0,5"/>

<StackPanel Grid.Row="13" Grid.ColumnSpan="2" Orientation="Horizontal"

HorizontalAlignment="Center" Margin="0,15,0,0">

<Button Content="Сохранить" Width="100" Margin="10,0" Click="SaveButton\_Click"/>

<Button Content="Отмена" Width="100" Margin="10,0" Click="CancelButton\_Click"/>

</StackPanel>

</Grid>', '<Grid Margin="20">

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="150"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Label Content="Название:" Grid.Row="0" Grid.Column="0" VerticalAlignment="Center"/>

<TextBox x:Name="NameTextBox" Grid.Row="0" Grid.Column="1" Margin="0,5"/>

<Label Content="Тип:" Grid.Row="1" Grid.Column="0" VerticalAlignment="Center"/>

<TextBox x:Name="TypeTextBox" Grid.Row="1" Grid.Column="1" Margin="0,5"/>

<Label Content="Описание:" Grid.Row="2" Grid.Column="0" VerticalAlignment="Center"/>

<TextBox x:Name="DescriptionTextBox" Grid.Row="2" Grid.Column="1" Margin="0,5"/>

<Label Content="Мин. цена:" Grid.Row="3" Grid.Column="0" VerticalAlignment="Center"/>

<TextBox x:Name="MinPartnerPriceTextBox" Grid.Row="3" Grid.Column="1" Margin="0,5"/>

<Label Content="Длина упаковки:" Grid.Row="4" Grid.Column="0" VerticalAlignment="Center"/>

<TextBox x:Name="PackageLengthTextBox" Grid.Row="4" Grid.Column="1" Margin="0,5"/>

<Label Content="Ширина упаковки:" Grid.Row="5" Grid.Column="0" VerticalAlignment="Center"/>

<TextBox x:Name="PackageWidthTextBox" Grid.Row="5" Grid.Column="1" Margin="0,5"/>

<Label Content="Высота упаковки:" Grid.Row="6" Grid.Column="0" VerticalAlignment="Center"/>

<TextBox x:Name="PackageHeightTextBox" Grid.Row="6" Grid.Column="1" Margin="0,5"/>

<Label Content="Вес без упаковки:" Grid.Row="7" Grid.Column="0" VerticalAlignment="Center"/>

<TextBox x:Name="WeightWithoutPackageTextBox" Grid.Row="7" Grid.Column="1" Margin="0,5"/>

<Label Content="Вес с упаковкой:" Grid.Row="8" Grid.Column="0" VerticalAlignment="Center"/>

<TextBox x:Name="WeightWithPackageTextBox" Grid.Row="8" Grid.Column="1" Margin="0,5"/>

<Label Content="Номер стандарта:" Grid.Row="9" Grid.Column="0" VerticalAlignment="Center"/>

<TextBox x:Name="StandardNumberTextBox" Grid.Row="9" Grid.Column="1" Margin="0,5"/>

<Label Content="Себестоимость:" Grid.Row="10" Grid.Column="0" VerticalAlignment="Center"/>

<TextBox x:Name="CostPriceTextBox" Grid.Row="10" Grid.Column="1" Margin="0,5"/>

<Button Content="Сохранить" Grid.Row="13" Grid.ColumnSpan="2" Width="150" Height="35"

Margin="0,15,0,0" HorizontalAlignment="Center" Click="SaveButton\_Click"/>

</Grid>');

INSERT INTO Programmists (PartnerTypeId, CompanyName, LegalAddress, INN, Phone, Email) VALUES

(' public partial class MebelEntities : DbContext

{

private static MebelEntities \_context;

public MebelEntities()

: base("name=MebelEntities")

{

}

public static MebelEntities GetContext()

{

if (\_context == null)

\_context = new MebelEntities();

return \_context;

}', 'public partial class AuthWindow : Window

{

MebelEntities db;

MainWindow mw;

ManagerWindow ManagerWindow;

public AuthWindow()

{

InitializeComponent();

db = new MebelEntities();

mw = new MainWindow();

ManagerWindow = new ManagerWindow();

}

private void LoginButton\_Click(object sender, RoutedEventArgs e)

{

var db = new MebelEntities();

string login = LogTB.Text.Trim();

string password = PaswwordTB.Password;

var user = db.Users

.Include("Roles")

.FirstOrDefault(u => u.Username == login && u.Password == password);

if (user != null)

{

switch (user.Roles.Name)

{

case "Manager":

new ManagerWindow().Show();

break;

case "Employee":

new MainWindow().Show();

break;

case "Admin":

new MainWindow().Show();

break;

}

this.Close();

}

else

{

MessageBox.Show("Неверный логин или пароль", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}', 'public partial class MainWindow : Window

{

MebelEntities db = new MebelEntities();

public MainWindow()

{

InitializeComponent();

LoadProducts();

}

private void LoadProducts()

{

// Загружаем список продуктов из базы

ProductGrid.ItemsSource = db.Products.ToList();

}

private void AddButton\_Click(object sender, RoutedEventArgs e)

{

AddProductWindow addWindow = new AddProductWindow(db);

addWindow.ShowDialog();

LoadProducts();

}

private void EditButton\_Click(object sender, RoutedEventArgs e)

{

if (ProductGrid.SelectedItem is Products product)

{

EditProductWindow editWindow = new EditProductWindow(product, db);

editWindow.ShowDialog();

LoadProducts();

}

else

{

MessageBox.Show("Выберите продукт для редактирования.");

}

}

}', 'public partial class ManagerWindow : Window

{

private MebelEntities db;

public ManagerWindow()

{

InitializeComponent();

db = new MebelEntities();

LoadProducts();

}

private void LoadProducts()

{

// Загружаем список продуктов из базы

ProductGrid.ItemsSource = db.Products.ToList();

}

private void RefreshButton\_Click(object sender, RoutedEventArgs e)

{

LoadProducts();

}

private void LogoutButton\_Click(object sender, RoutedEventArgs e)

{

AuthWindow authWindow = new AuthWindow();

authWindow.Show();

this.Close();

}

}', ' public partial class AddProductWindow : Window

{

private MebelEntities \_db;

public AddProductWindow(MebelEntities db)

{

InitializeComponent();

\_db = db;

}

private void SaveButton\_Click(object sender, RoutedEventArgs e)

{

try

{

var newProduct = new Products

{

Name = NameTextBox.Text,

Type = TypeTextBox.Text,

Description = DescriptionTextBox.Text,

MinPartnerPrice = int.TryParse(MinPartnerPriceTextBox.Text, out int minPrice) ? minPrice : 0,

PackageLength = int.TryParse(PackageLengthTextBox.Text, out int length) ? length : 0,

PackageWidth = int.TryParse(PackageWidthTextBox.Text, out int width) ? width : 0,

PackageHeight = int.TryParse(PackageHeightTextBox.Text, out int height) ? height : 0,

WeightWithoutPackage = int.TryParse(WeightWithoutPackageTextBox.Text, out int wOut) ? wOut : 0,

WeightWithPackage = int.TryParse(WeightWithPackageTextBox.Text, out int wIn) ? wIn : 0,

StandardNumber = StandardNumberTextBox.Text,

CostPrice = int.TryParse(CostPriceTextBox.Text, out int cost) ? cost : 0,

};

\_db.Products.Add(newProduct);

\_db.SaveChanges();

MessageBox.Show("Продукт успешно добавлен.", "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

this.Close();

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка при добавлении продукта: {ex.Message}", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}', 'public partial class EditProductWindow : Window

{

private Products \_product;

private MebelEntities \_db;

public EditProductWindow(Products product, MebelEntities db)

{

InitializeComponent();

\_db = db;

\_product = product;

NameTextBox.Text = \_product.Name;

TypeTextBox.Text = \_product.Type;

DescriptionTextBox.Text = \_product.Description;

MinPartnerPriceTextBox.Text = \_product.MinPartnerPrice.ToString();

PackageLengthTextBox.Text = \_product.PackageLength.ToString();

PackageWidthTextBox.Text = \_product.PackageWidth.ToString();

PackageHeightTextBox.Text = \_product.PackageHeight.ToString();

WeightWithoutPackageTextBox.Text = \_product.WeightWithoutPackage.ToString();

WeightWithPackageTextBox.Text = \_product.WeightWithPackage.ToString();

StandardNumberTextBox.Text = \_product.StandardNumber;

CostPriceTextBox.Text = \_product.CostPrice.ToString();

}

private void SaveButton\_Click(object sender, RoutedEventArgs e)

{

if (string.IsNullOrWhiteSpace(NameTextBox.Text) ||

string.IsNullOrWhiteSpace(TypeTextBox.Text) ||

!int.TryParse(MinPartnerPriceTextBox.Text, out int minPrice))

{

MessageBox.Show("Пожалуйста, заполните все обязательные поля корректно.",

"Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

\_product.Name = NameTextBox.Text.Trim();

\_product.Type = TypeTextBox.Text.Trim();

\_product.Description = DescriptionTextBox.Text.Trim();

\_product.MinPartnerPrice = minPrice;

\_product.PackageLength = int.TryParse(PackageLengthTextBox.Text, out int length) ? length : 0;

\_product.PackageWidth = int.TryParse(PackageWidthTextBox.Text, out int width) ? width : 0;

\_product.PackageHeight = int.TryParse(PackageHeightTextBox.Text, out int height) ? height : 0;

\_product.WeightWithoutPackage = int.TryParse(WeightWithoutPackageTextBox.Text, out int wOut) ? wOut : 0;

\_product.WeightWithPackage = int.TryParse(WeightWithPackageTextBox.Text, out int wIn) ? wIn : 0;

\_product.StandardNumber = StandardNumberTextBox.Text.Trim();

\_product.CostPrice = int.TryParse(CostPriceTextBox.Text, out int cost) ? cost : 0;

try

{

\_db.SaveChanges();

MessageBox.Show("Данные успешно обновлены.", "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

this.DialogResult = true;

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка при сохранении: {ex.Message}", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void CancelButton\_Click(object sender, RoutedEventArgs e)

{

this.DialogResult = false;

}

}'

);