

Рисунок 1 - Форма авторизации

Все скриншоты далее будут представлены с условием входа в систему, как администратора

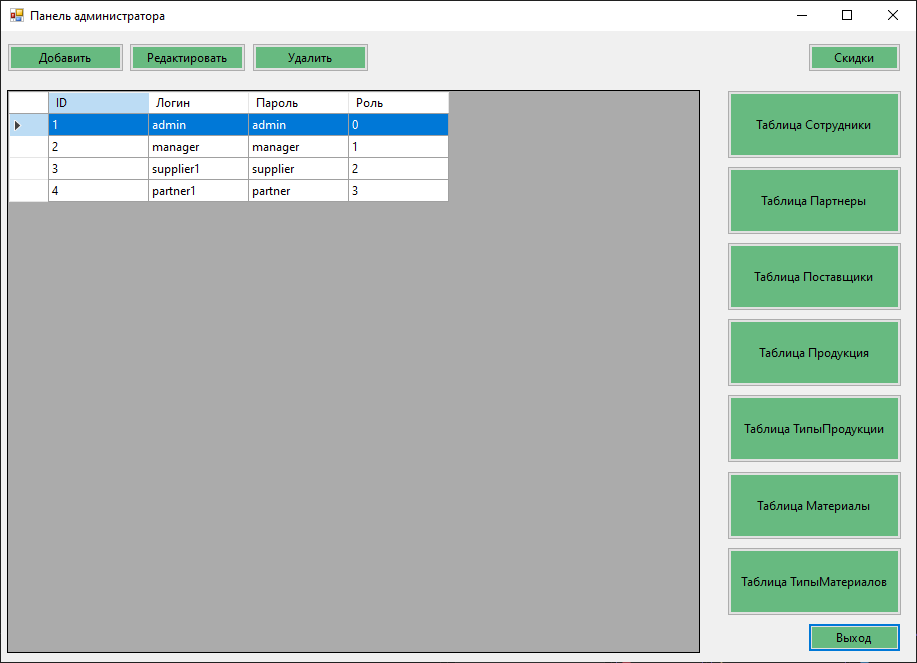


Рисунок 2 - Главная форма администратора

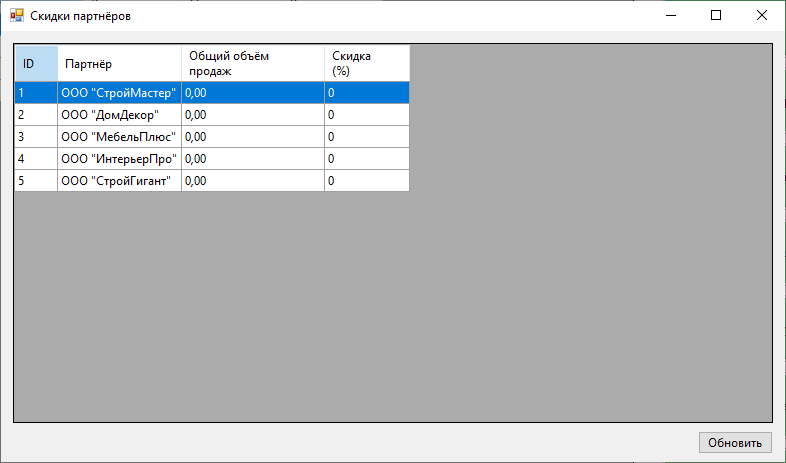


Рисунок 3 - Таблица скидок партнеров

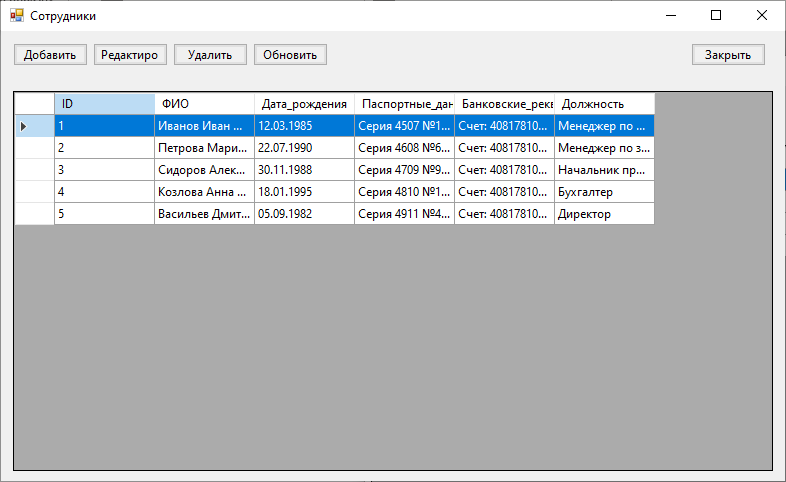


Рисунок 4 - Таблица сотрудников

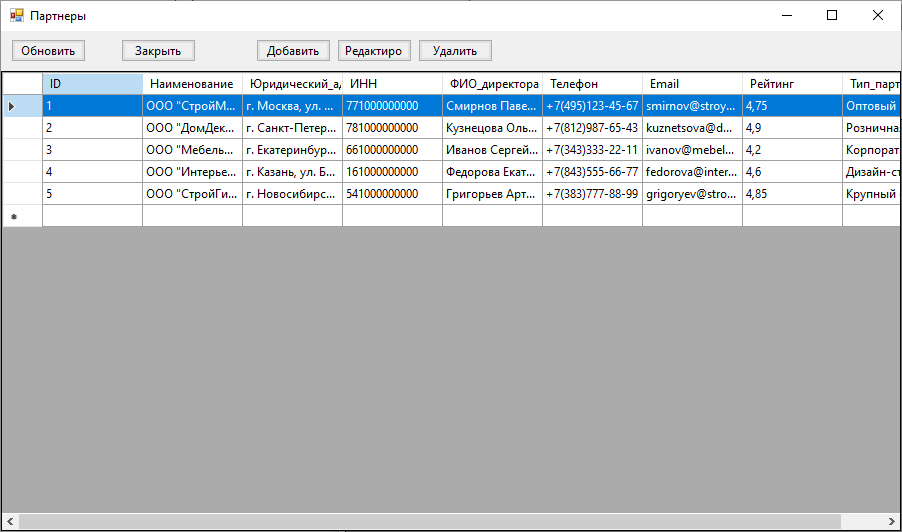


Рисунок 5 - Таблица партнеров

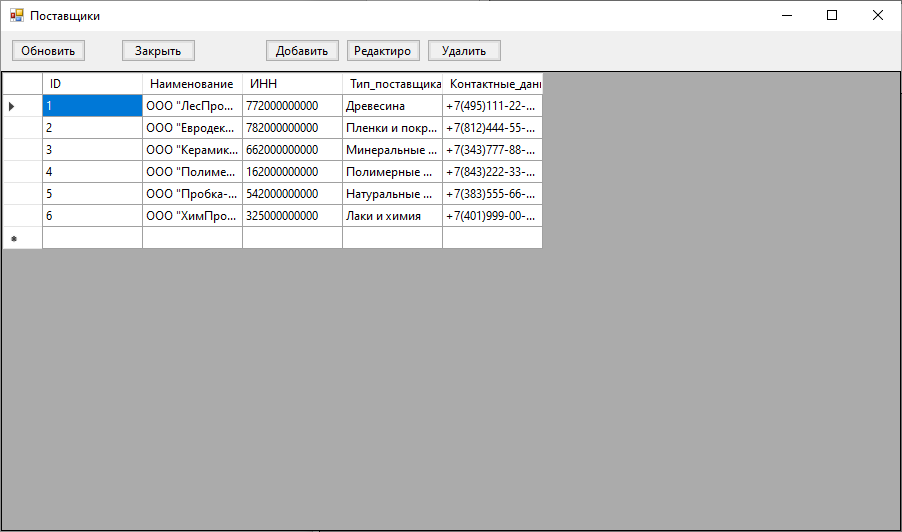


Рисунок 6 - Таблица поставщиков

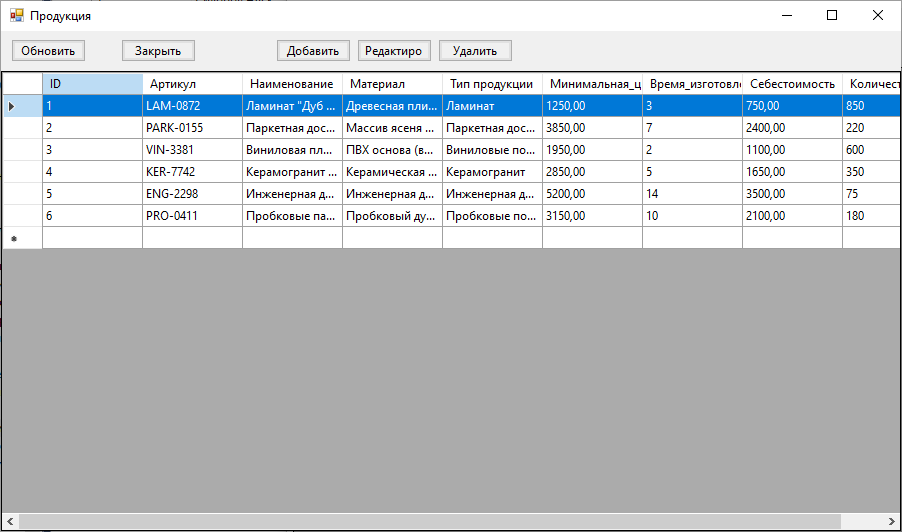


Рисунок 7 - Таблица продукция

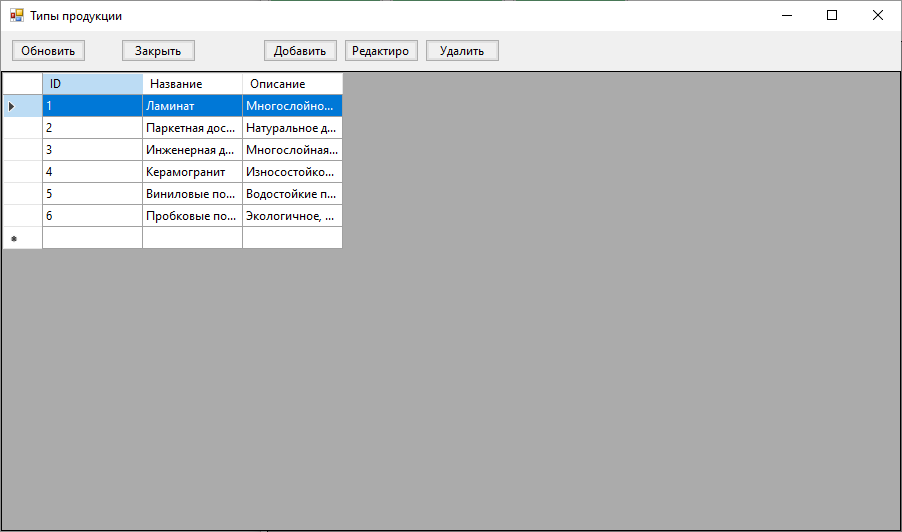


Рисунок 8 - Таблица типы продукции

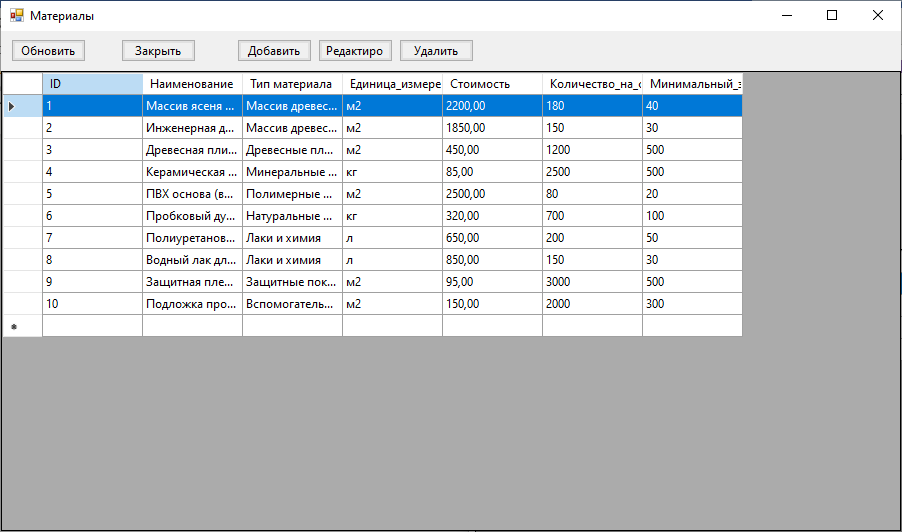


Рисунок 9 - Таблица материалы

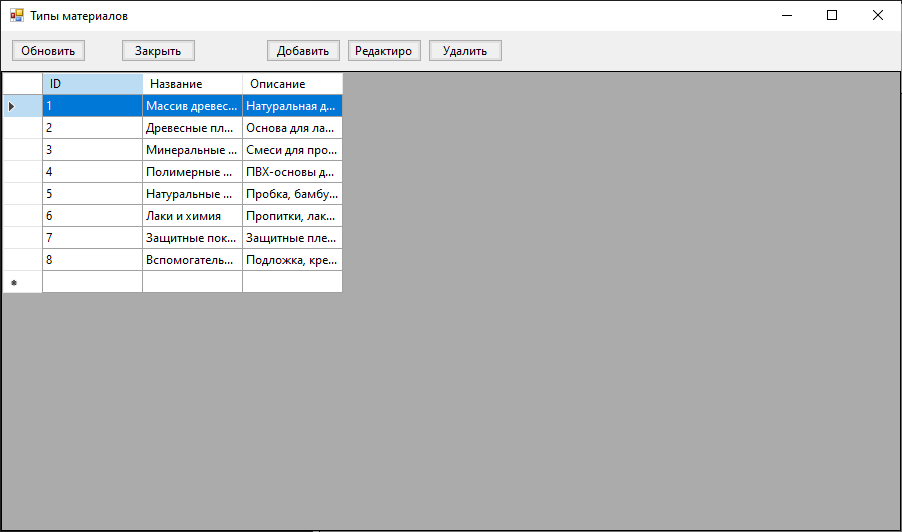


Рисунок 10 - Таблица типы материалов

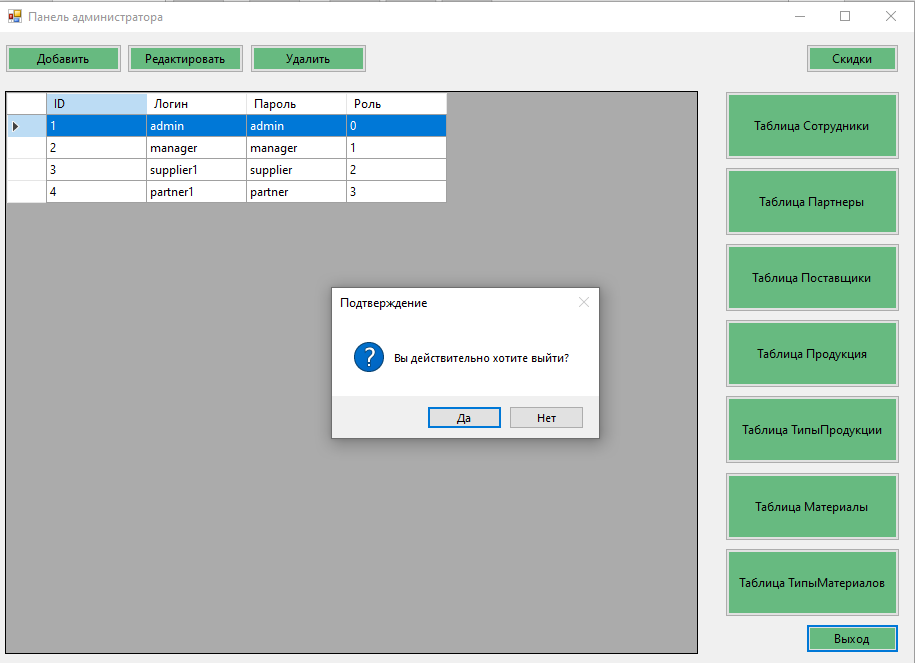


Рисунок 11 - есть возможность вернуться к форме авторизации

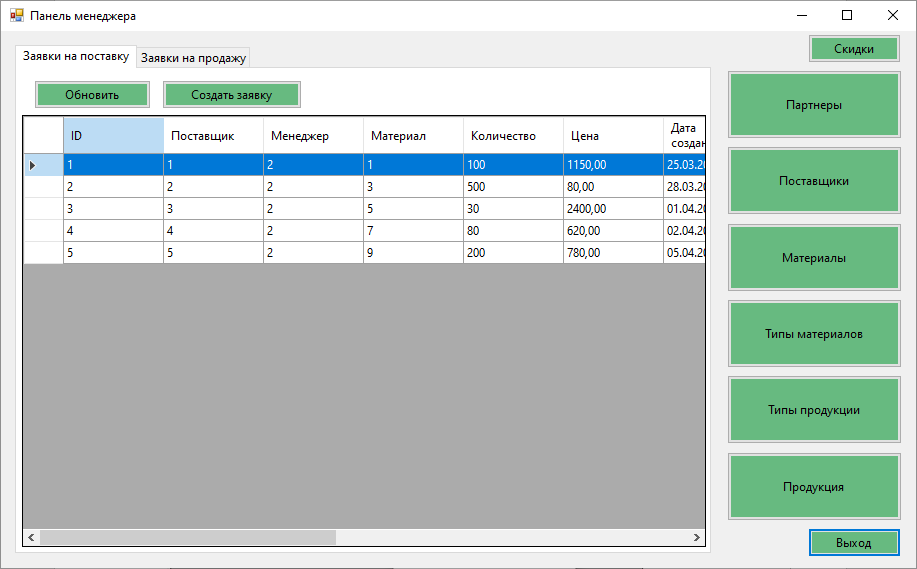


Рисунок 12 - Главная форма менеджера

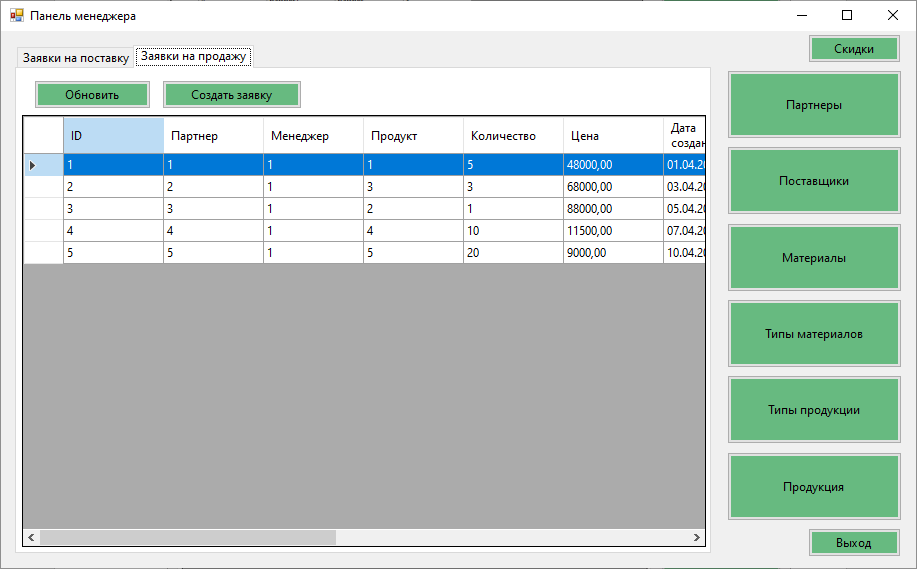


Рисунок 13 - Главная форма менеджера

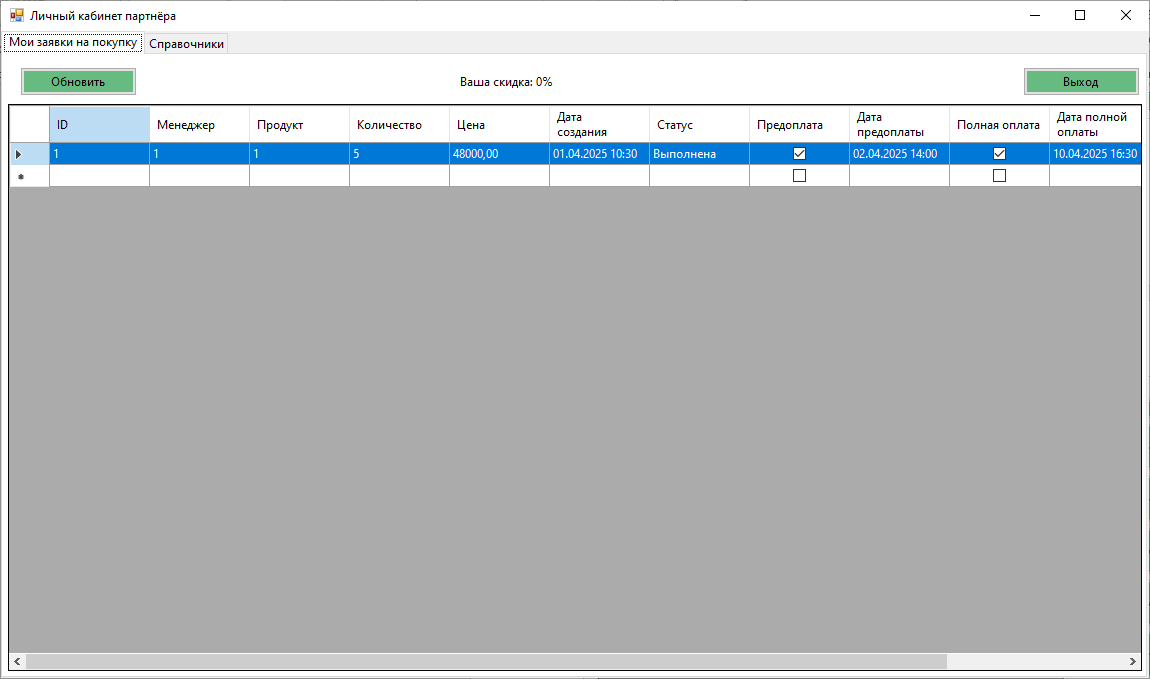


Рисунок 14 - Главная форма партнера

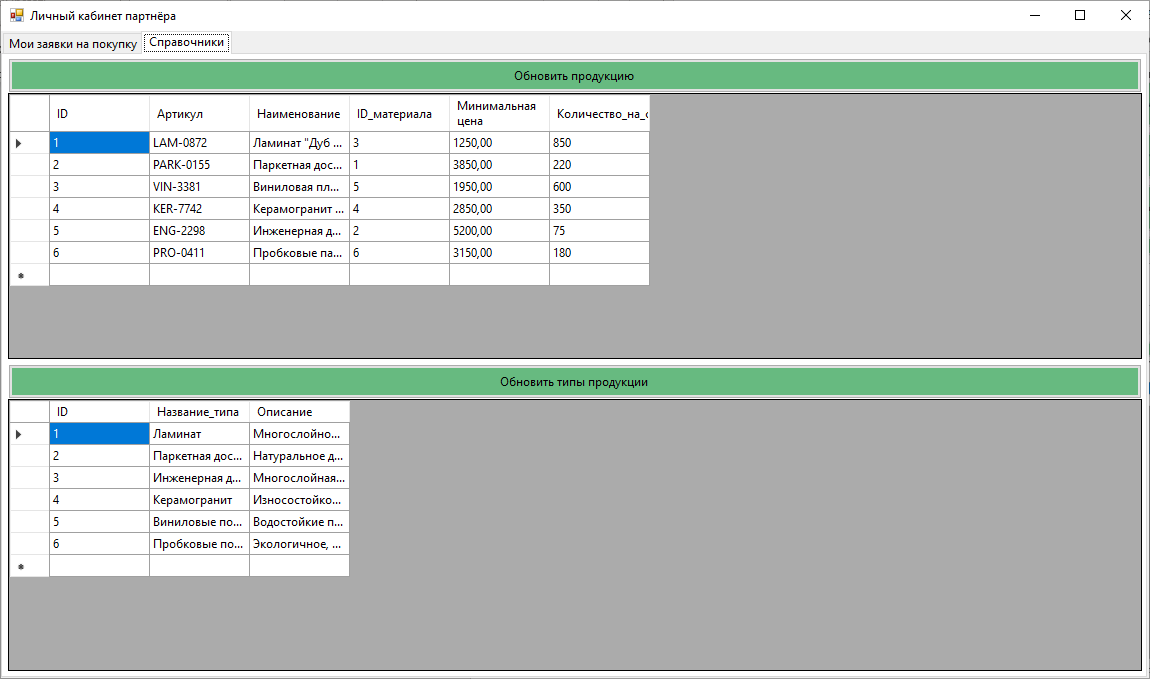


Рисунок 15 - Главная форма партнера

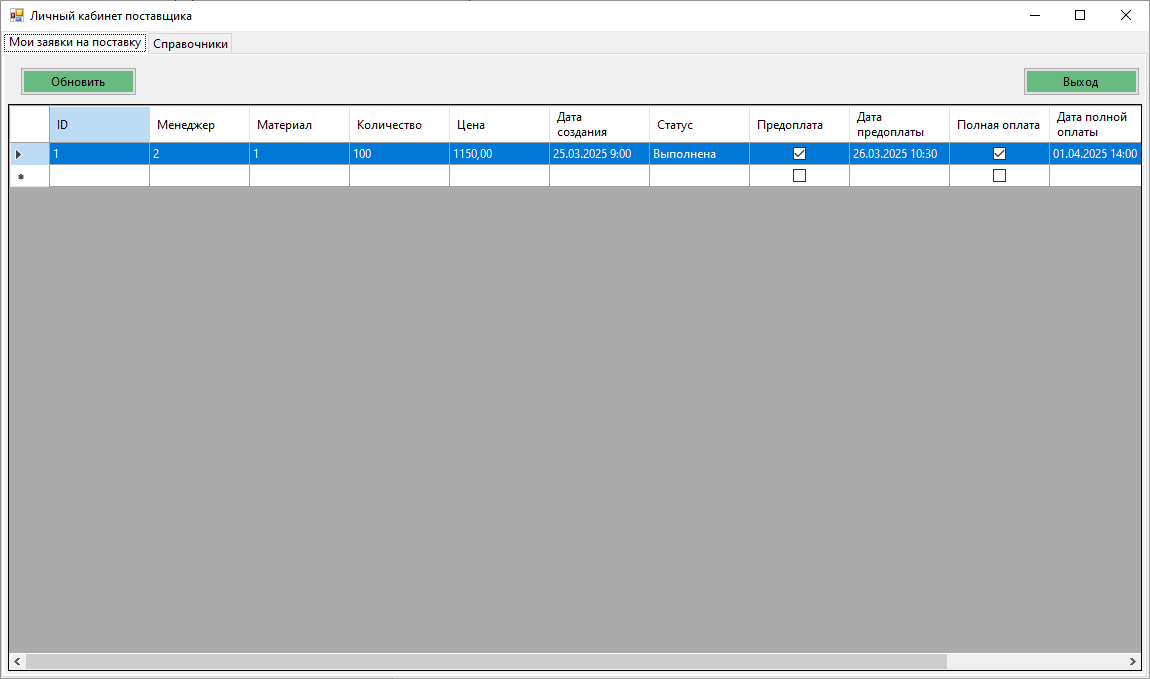


Рисунок 16 - Главная форма поставщика

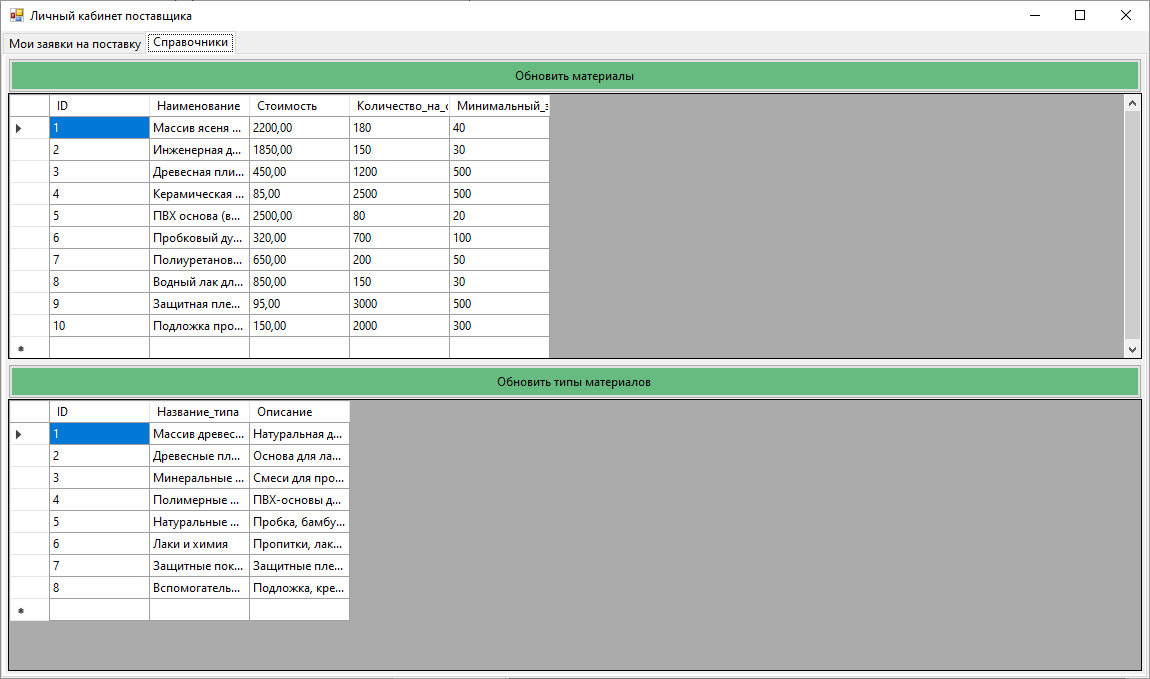


Рисунок 17 - Главная форма поставщика

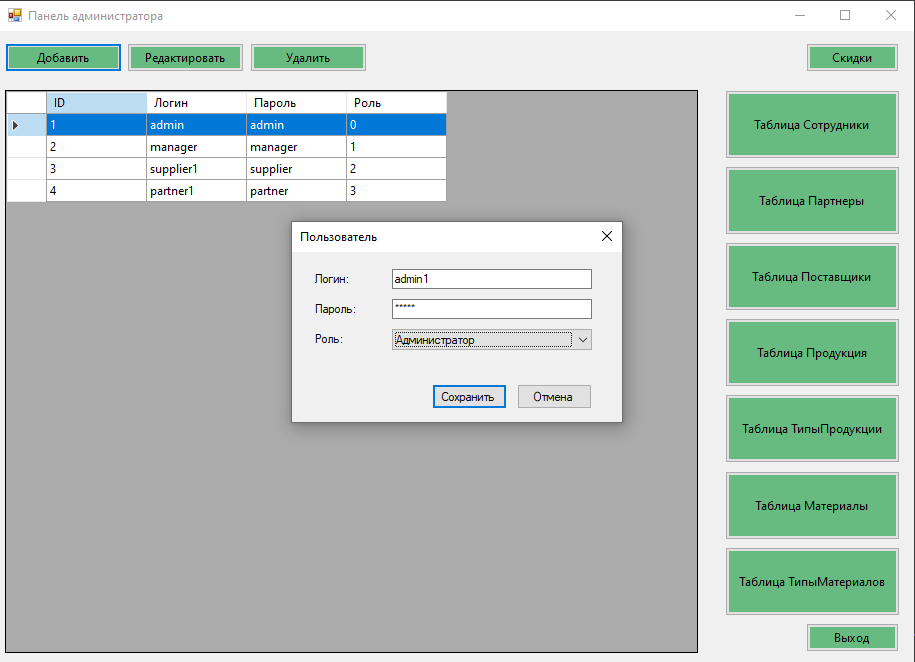


Рисунок 18 - Добавление нового пользователя

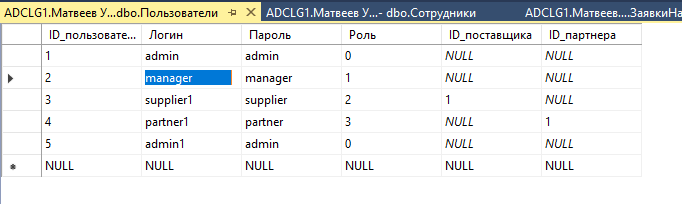


Рисунок 19 - Появившийся пользователь в БД

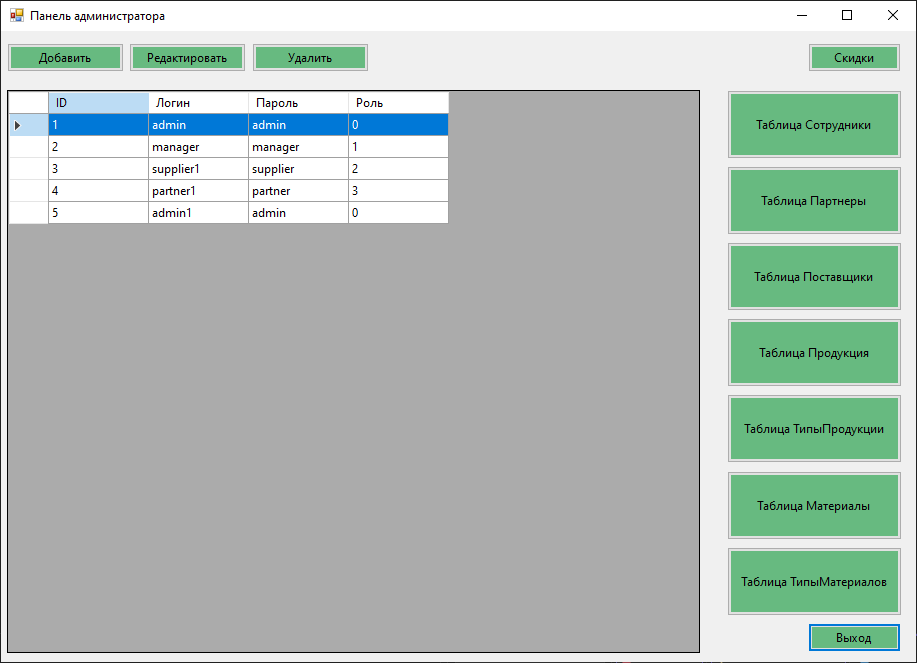


Рисунок 20 - Обновление таблицы из Бд в форме

Метод для расчета количества материала:

public static class ProductionCalculator

{

public static int CalculateRequiredMaterial(

int idProductType,

int idMaterialType,

int productQuantity,

double param1,

double param2)

{

// Валидация входных данных

if (productQuantity <= 0 || param1 <= 0 || param2 <= 0)

return -1;

string connStr = ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString;

double productCoefficient = 0;

double wastePercentage = 0;

try

{

using (SqlConnection conn = new SqlConnection(connStr))

{

conn.Open();

// Получаем коэффициент типа продукции

string prodQuery = "SELECT Коэффициент FROM ТипыПродукции WHERE ID\_типа\_продукции = @id";

using (SqlCommand cmd = new SqlCommand(prodQuery, conn))

{

cmd.Parameters.AddWithValue("@id", idProductType);

var result = cmd.ExecuteScalar();

if (result == null || result == DBNull.Value)

return -1;

productCoefficient = Convert.ToDouble(result);

}

// Получаем процент брака материала

string matQuery = "SELECT Процент\_брака FROM ТипыМатериалов WHERE ID\_типа\_материала = @id";

using (SqlCommand cmd = new SqlCommand(matQuery, conn))

{

cmd.Parameters.AddWithValue("@id", idMaterialType);

var result = cmd.ExecuteScalar();

if (result == null || result == DBNull.Value)

return -1;

wastePercentage = Convert.ToDouble(result);

}

}

// Расчёт

double materialPerUnit = param1 \* param2 \* productCoefficient;

double totalMaterial = materialPerUnit \* productQuantity;

// Учёт брака: если брак 5%, то нужно на 5% больше

double totalWithWaste = totalMaterial \* (1 + wastePercentage / 100.0);

// Округляем вверх до целого (нельзя использовать дробный материал)

int resultQuantity = (int)Math.Ceiling(totalWithWaste);

return resultQuantity;

}

catch

{

return -1;

}

}

}