

**ОТЧЕТ**

по учебной практике

**УП.02.01. «Разработка и администрирование баз данных»**

по специальности 09.02.03 «Программирование в компьютерных системах»

Выполнил студент гр. П1-18

Курбанмухаммедов Стив Джумамурадович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (подпись)

Проверил преподаватель

Гусятинер Л. Б.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(оценка)

Королев, 2021

**Оглавление**

[1. Проектирование базы данных 2](#_Toc87873620)

[1.1 Построение ER-модели базы данных 2](#_Toc87873621)

[2. Разработка приложения 2](#_Toc87873622)

[2.1 Разработка таблиц и связей 2](#_Toc87873623)

[2.2 Разработка хранимых процедур 4](#_Toc87873624)

[2.3 Разработка клиентского приложения 4](#_Toc87873625)

[Заключение 8](#_Toc87873626)

[Список литературы и интернет ресурсов 9](#_Toc87873627)

# Проектирование базы данных

## Построение ER-модели базы данных

В результате изучения предметной области и существующих форм данных была построена ER-модель базы данных для приложения.

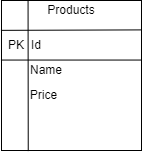


Рисунок 1. ER-модель базы данных

# Разработка приложения

## Разработка таблиц и связей

В программе “Интернет магазин для покупки электроники” предназначенная для разработки и администрирования баз данных SQL, были созданы таблицы и связи, которые были спроектированы в ER-модели.

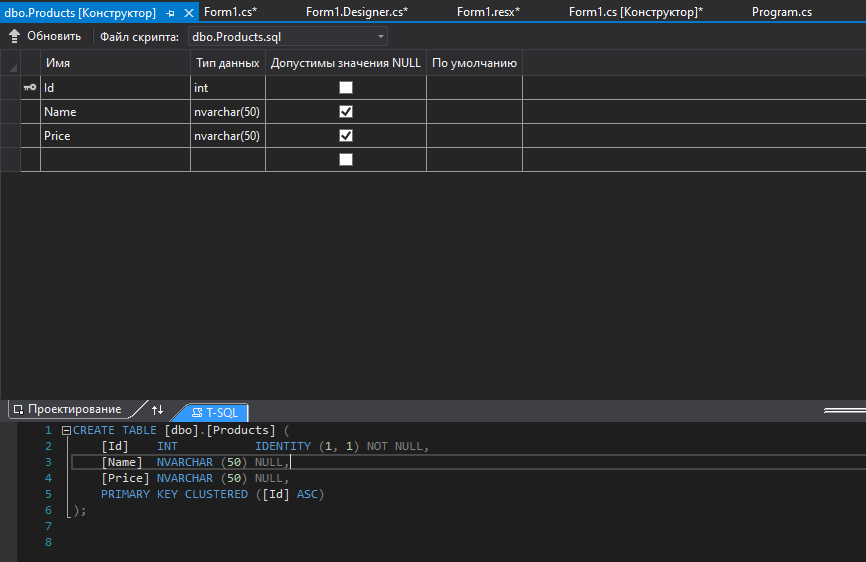


Рисунок 2. Таблица Document

## Разработка хранимых процедур

После создания таблиц была создана хранимая процедура для добавления профессии в таблицу dbo.Products.

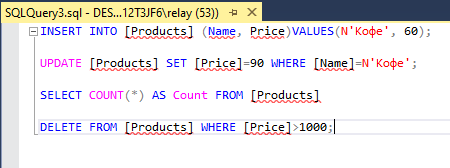


Рисунок 7. Хранимая процедура SQLQuery3.sql

## Разработка клиентского приложения

Клиентское приложение было разработано на языке программирования C#.

Для того чтобы подключиться к БД, надо лишь вызвать класс Form1.cs и передать ему как аргумент имя БД, к которой надо подключаться.

Класс SQL выполняет следующие функции:

1. Подключение к БД
2. Подключение к таблице
3. Получение ID записи
4. Добавление записи в таблицу
5. Удаление записи из таблицы
6. Обновление записи в таблице

Листинг 1. Функция подключения к базе данных.

private async void Form1\_Load(object sender, EventArgs e)

{

string connectionString = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\relay\OneDrive\Рабочий стол\Курсач БД\WorkingWithDB\WorkingWithDB\Database.mdf;Integrated Security=True";

sqlConnection = new SqlConnection(connectionString);

await sqlConnection.OpenAsync();

SqlDataReader sqlReader = null;

SqlCommand command = new SqlCommand("SELECT \* FROM [Products]", sqlConnection);

try

{

sqlReader = await command.ExecuteReaderAsync();

while (await sqlReader.ReadAsync())

{

listBox1.Items.Add(Convert.ToString(sqlReader["Id"]) + " " + Convert.ToString(sqlReader["Name"]) + " " + Convert.ToString(sqlReader["Price"]));

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString(), ex.Source.ToString(), MessageBoxButtons.OK, MessageBoxIcon.Error);

}

finally

{

if (sqlReader != null)

sqlReader.Close();

}

}

Листинг 2. Функция обновления базы данных.

private async void обновитьToolStripMenuItem\_Click(object sender, EventArgs e)

{

listBox1.Items.Clear();

SqlDataReader sqlReader = null;

SqlCommand command = new SqlCommand("SELECT \* FROM [Products]", sqlConnection);

try

{

sqlReader = await command.ExecuteReaderAsync();

while (await sqlReader.ReadAsync())

{

listBox1.Items.Add(Convert.ToString(sqlReader["Id"]) + " " + Convert.ToString(sqlReader["Name"]) + " " + Convert.ToString(sqlReader["Price"]));

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString(), ex.Source.ToString(), MessageBoxButtons.OK, MessageBoxIcon.Error);

}

finally

{

if (sqlReader != null)

sqlReader.Close();

}

}

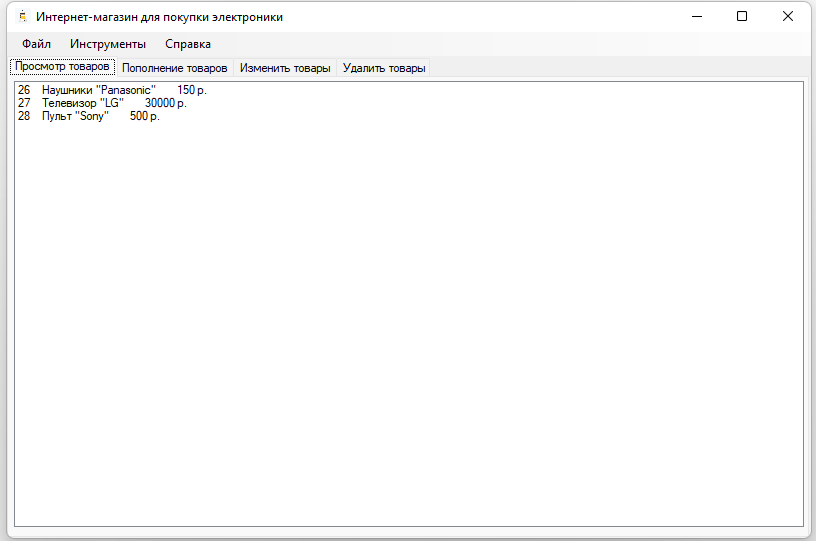


Рисунок 9. Клиентское приложение

# Заключение

В результате учебной практики была спроектирована и создана база данных. Также было создано клиентское приложение для использования данной базы данных.

# Список литературы и интернет ресурсов

1. IBExpert

<https://www.ibexpert.net>

1. Firebird

<https://firebirdsql.org>

1. Python

<https://www.python.org/>