

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет _____ Информационных технологий
Кафедра _____ Информационные системы и технологии
Специальность _____ 1–40 01 01 Программное обеспечение информационных
технологий
Специализация _____ 1 –40 01 01 Программное обеспечение информационных
технологий (программирование интернет – изданий)

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К КУРСОВОМУ ПРОЕКТУ НА ТЕМУ:**

«Сеть супермаркетов»

Выполнил студент _____ Матюх Арсений Артурович
(Ф.И.О.)

Руководитель проекта _____ пр.-стаж. Нистюк О.А.
(учен. степень, звание, должность, Ф.И.О., подпись)

Заведующий кафедрой _____ к.т.н., доц. Смелов В.В.
(учен. степень, звание, должность, Ф.И.О., подпись)

Консультант: _____ пр.-стаж. Нистюк О.А.
(учен. степень, звание, должность, Ф.И.О., подпись)

Нормоконтролер: _____ пр.-стаж. Нистюк О.А.
(учен. степень, звание, должность, Ф.И.О., подпись)

Курсовой проект защищен с оценкой _____

Реферат

Пояснительная записка курсового проекта содержит 30 страниц пояснительной записки, 27 иллюстраций, 6 источников литературы, 3 приложения.

NODE.JS, JAVASCRIPT, MICROSOFT SQL SERVER, EXPRESS.JS, NODE-MSSQL, MSNODESQLV8, BACKUP DATABASE, RESTORE DATABASE

Основная цель курсового проекта: проектирование базы данных для сети магазинов и технология резервного копирования и восстановления базы данных.

В первой главе проводится аналитический обзор литературы по тематике курсового проекта и содержит анализ методов решения поставленных задач.

Вторая глава посвящена процессу проектирования системы.

В третьей главе приведено обоснование технических приёмов.

В четвертой главе описано тестирование, позволяющее подробно понять интерфейс программного средства.

В заключении приведены результаты проделанной работы.

				КП 00.00.ПЗ						
	ФИО	Подпись	Дата							
Разраб.	Матюх А.А.			Реферат			Лит.	Лист	Листов	
Провер.	Нистюк О.А.							1	1	
							БГТУ 71171075, 2019			
Н. контр.	Нистюк О.А.									
Утверд.	Нистюк О.А.									

Abstract

The explanatory note of the course project contains 30 pages of the explanatory note, 27 illustrations, 6 sources of literature, 3 appendices.

NODE.JS, JAVASCRIPT, MICROSOFT SQL SERVER, EXPRESS.JS, NODE-MSSQL, MSNODESQLV8, BACKUP DATABASE, RESTORE DATABASE

The main goal of the course project: designing a database for a chain of stores and technology for backing up and restoring a database.

The first chapter provides an analytical review of the literature on the subject of the course project and contains an analysis of methods for solving the tasks.

The second chapter is devoted to the system design process.

The third chapter provides the rationale for technical techniques.

The fourth chapter describes testing, which allows you to understand in detail the interface of a software tool.

In conclusion, the results of the work done are presented.

				КП 00.00.ПЗ						
	ФИО	Подпись	Дата							
Разраб.	Матюх А.А.			Abstract			Лит.	Лист	Листов	
Провер.	Нистюк О.А.								1	1
							БГТУ 71171075, 2019			
Н. контр.	Нистюк О.А.									
Утверд.	Нистюк О.А.									

Содержание

Введение	7
1 Аналитический обзор литературы	8
2 Разработка программной системы	11
2.1 Разработка модели базы данных	11
2.2 Хранимые процедуры.....	14
2.2.1 Выборка данных для таблиц.....	14
2.2.2 Выборка данных для dropdown полей	14
2.2.3 Выборка данных по поисковому запросу	14
2.2.4 Заполнение таблиц 100 000 строк	15
2.2.5 Вход администратора в аккаунт.....	15
2.2.6 Удаление данных из таблиц	15
2.2.7 Добавление данных в таблицы.....	15
2.2.8 Изменение данных в таблицах	16
2.2.9 Экспорт и импорт таблицы PRODUCTS в формате xml	16
2.2.10 Резервное копирование и восстановление базы данных	17
2.3 Индексы	18
3 Обоснование технических приемов программирования	19
4 Тестирование, проверка работоспособности и анализ данных.....	21
Заключение	29
Список литературы	30
Приложение А	31
Приложение Б.....	32
Приложение В	33

				КП 00.00.ПЗ						
	ФИО	Подпись	Дата							
Разраб.	Матюх А.А.			Содержание			Лит.	Лист	Листов	
Провер.	Нистюк О.А.								1	1
							БГТУ 71171075, 2019			
Н. контр.	Нистюк О.А.									
Утверд.	Нистюк О.А.									

Введение

Целью данной работы была разработка реляционной базы данных на тему «Сеть супермаркетов». База данных должна быть составлена для работы торговых сетей. Так же было необходимо разработать соответствующее приложение, для демонстрации её работы.

База данных — это организованная структура, предназначенная для хранения информации, систематизированная таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины.

Реляционная база данных — база данных, основанная на реляционной модели данных.

В качестве СУБД для базы данных была выбрана Microsoft SQL Server, в связи с ее простотой, производительностью и надежностью.

Так же было необходимо создать программу для демонстрации работы базы данных. Написанное приложение строится на клиент-серверной архитектуре. В качестве клиента выступает приложение написанное на HTML и JavaScript, которое отображается в браузере. Для сервера была выбрана платформа Node.js, в связи с кроссплатформенностью, простотой развертки сервера и гибкости работы за счёт npm.

Node.js — программная платформа, основанная на движке V8. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API, подключать другие внешние библиотеки. Node.js применяется преимущественно на сервере. В основе Node.js лежит событийно-ориентированное и асинхронное программирование.

Npm — менеджер пакетов, входящий в состав Node.js, позволяющий расширить возможности приложения.

Для работы с Node.js были выбраны следующие пакеты:

- Express.js — предоставляет собой слой фундаментальных функций для упрощения разработки веб-приложений;
- Body-parser — пакет для парсинга body;
- Msnodsqlv8 — TDS драйвер для работы с Microsoft SQL Server, основан на драйвере node-mssql;
- Node-mssql — Node.js драйвер клиента для Microsoft SQL Server;
- Fs — стандартный пакет Node.js для работы с вводом-выводом файловой системы.

				КП 00.00.ПЗ			
	ФИО	Подпись	Дата				
Разраб.	Матюх А.А.			Введение		Лит.	Лист
Провер.	Нистюк О.А.						Листов
						1	1
Н. контр.	Нистюк О.А.					БГТУ 71171075, 2019	
Утверд.	Нистюк О.А.						

1 Аналитический обзор литературы

Раньше, для администрирования магазинов, вели бухгалтерский учет, что осложняло весь процесс. Сейчас же используются базы данных с разработанным для них программным обеспечением, позволяя уменьшить объем бумажных работ и оптимизировать работу всей системы магазинов, попутно увеличивая прибыль компании, использующей современные технологии.

Постепенно с развитием программного обеспечения ЭВМ появились идеи создания управляющих систем, которые позволяли бы накапливать, хранить и обновлять взаимосвязанные данные по целому комплексу решаемых задач, например, при автоматизации бухгалтерского учета на предприятии. Эти идеи нашли свое воплощение в системах управления базами данных (СУБД). СУБД взаимодействуют не с локальными, а взаимосвязанными по информации массивами, называемыми базами данных. С появлением персональных компьютеров СУБД становятся наиболее популярным средством обработки табличной информации. Они являются инструментальным средством проектирования банков данных при обработке больших объемов информации.

Базы данных, написанные для различных сетей супермаркетов, используются для работы за кассовым оборудованием и управления непосредственно базой от лица администраторов магазинов. В открытом доступе рассмотреть напрямую системы управления магазинами невозможно, так они пишутся на заказ. Но можно рассмотреть структуру организации товаров среди онлайн-магазинов ориентированные на продукты питания. Одним из таких белорусских магазинов таковых является «e-dostavka.by», принадлежащий ООО «Евроторг».

Рассмотрим в качестве примера каталог товаров данного сайта на рисунке 1.1.

				КП 01.00.ПЗ			
	ФИО	Подпись	Дата				
Разраб.	Матюх А.А.			1 Аналитический обзор литературы		Лит.	Лист
Провер.	Нистюк О.А.						Листов
						1	3
Н. контр.	Нистюк О.А.					БГТУ 71171075, 2019	
Утверд.	Нистюк О.А.						

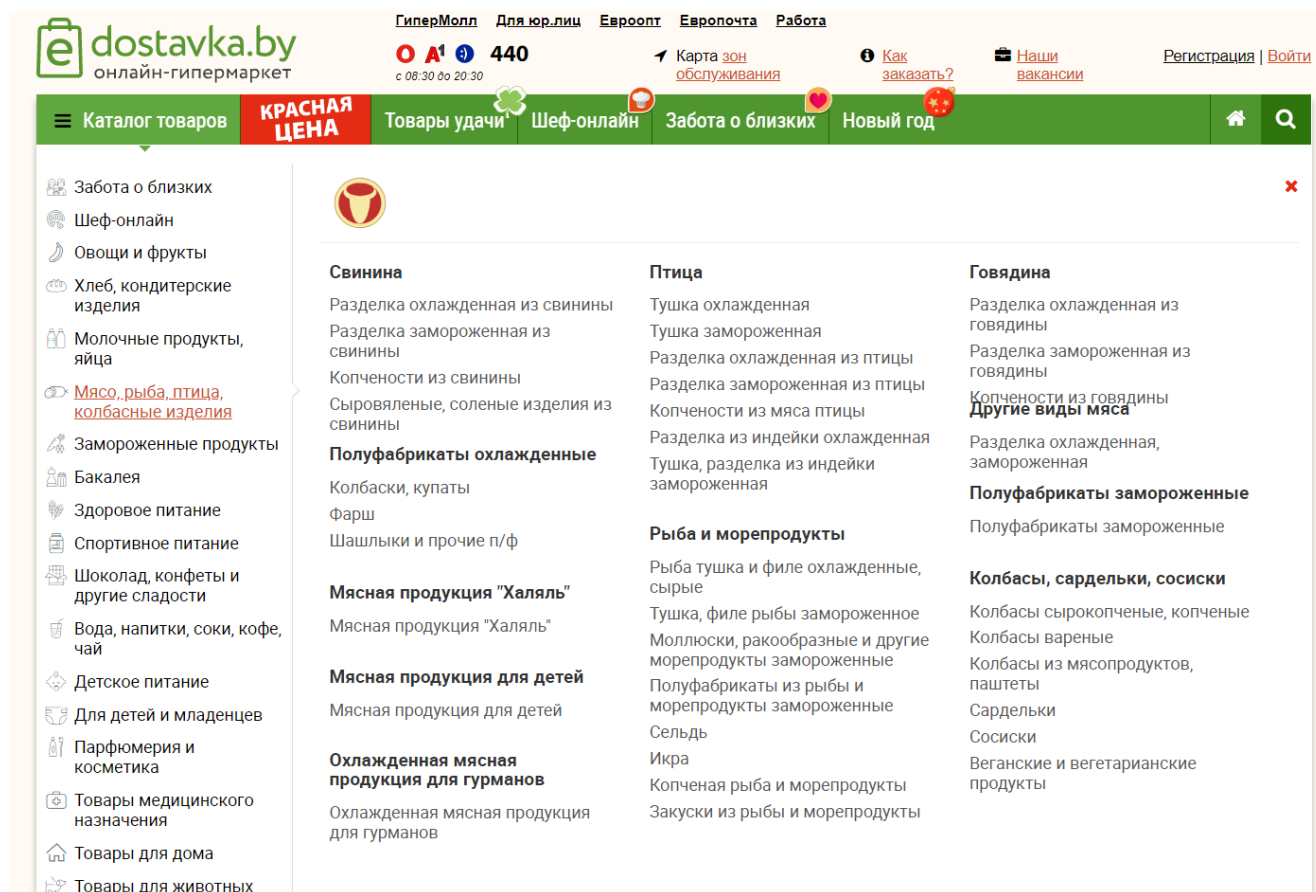


Рисунок 1.1 – Пример каталога «e-dostavka.by»

Тут структура каталога товаров делится на категории и подкатегории, и типы, а если опираться на структуру реальных магазинов, то это можно описать как деление на отделы, категории и типы продукции. У товара есть следующие особенности:

- Цена;
- Название;
- Масса или же объем;
- Артикул;
- Ссылки на отделы, категории и типы к которому относится товар;
- Возможность добавления в корзину.

Так же рассмотрим интернет-магазин натуральных продуктов «ecobar.by», одна из страниц которого представлена на рисунке 1.2.

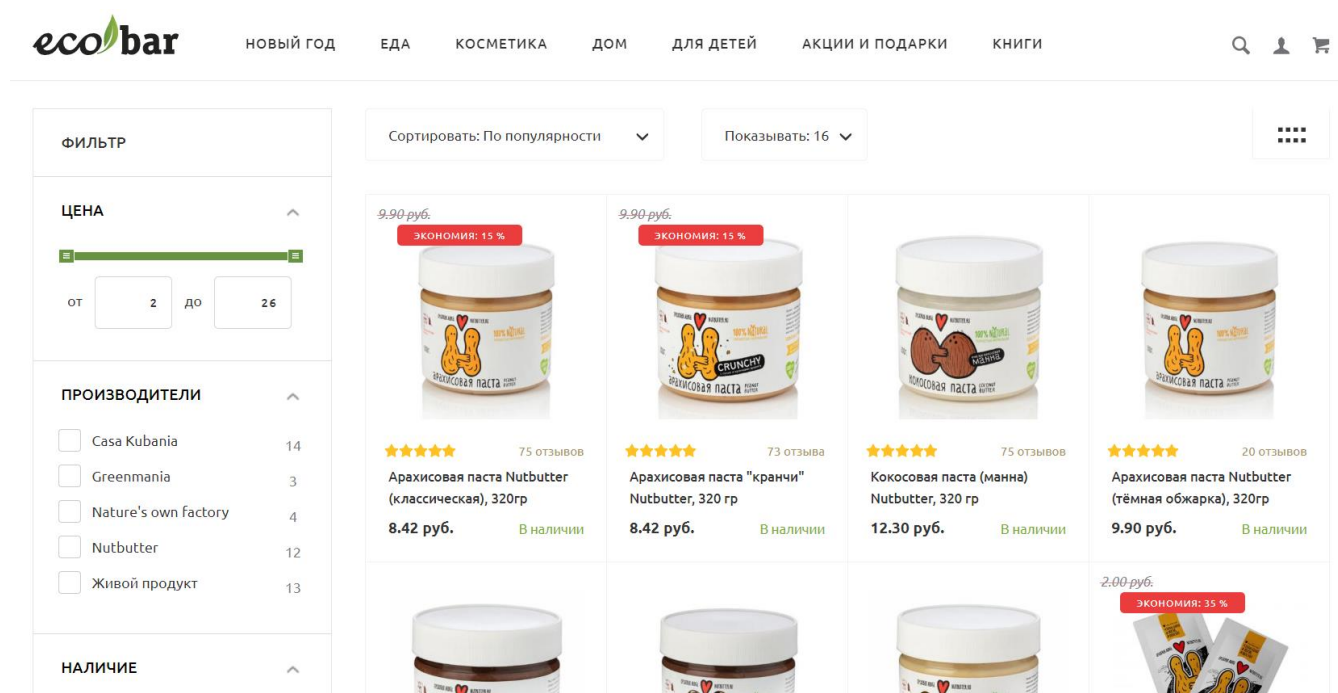


Рисунок 1.2 – Пример сайта «ecobar.by»

Его основные особенности такие же, как и у «e-dostavka.by», кроме одного, в характеристики включена вся информация с этикетки товара, такая как импортер, производитель, состав и другая.

Исходя из приведённых данных была разработана база данных, описанная в пункте 2.

2 Разработка программной системы

При разработке курсового проекта понадобились следующие объекты:

1. Таблицы;
2. Хранимые процедуры;
3. Индексы.

2.1 Разработка модели базы данных

Для реализации базы данных для сети магазинов было разработано 12 таблиц. Диаграмма базы со структурой связей представлена в приложении А.

Логически можно вывести 5 основных таблиц: STORES, EMPLOYEES, PRODUCTS, STORE_ADMIN и USERS.

Таблица STORES представляет магазин, в котором осуществляется продажа продукции, состоит из столбцов:

- Id_Store – идентификатор магазина, тип int, первичный ключ;
- Name_Store – название магазина, тип nvarchar;
- County – название страны, в которой расположен магазин, тип nvarchar;
- Town – название города, в котором расположен магазин, тип nvarchar;
- Adress – Адрес по которому расположен магазин, тип nvarchar;
- Id_Type – идентификатор типа магазина, тип int, внешний ключ;
- Size – площадь занимаемая магазином в кв. км, тип int.

Таблица EMPLOYEES представляет собой людей, работающих в магазинах, состоит из столбцов:

- Id_Empl – идентификатор сотрудника, тип int, первичный ключ;
- Name_First – имя, тип nvarchar;
- Name_Sec – фамилия, тип nvarchar;
- Name_Father – отчество, тип nvarchar;
- Sex – пол, тип nvarchar;
- Age – возраст, тип int;
- Exp – стаж, тип int;
- Id_Post – идентификатор должности работника, тип int, внешний ключ;
- Id_Store – идентификатор магазина, тип int, внешний ключ;
- Id_Dep – идентификатор отдела, тип int, внешний ключ.

Таблица USERS представляет собой покупателей магазина, состоит столбцов:

- Id_User – идентификатор покупателя, тип int, первичный ключ;
- Name_First – имя, тип nvarchar;
- Name_Sec – фамилия, тип nvarchar;

				КП 02.00.ПЗ						
	ФИО	Подпись	Дата							
Разраб.	Матюх А.А.			2 Разработка программной системы			Лит.	Лист	Листов	
Провер.	Нистюк О.А.								1	8
							БГТУ 71171075, 2019			
Н. контр.	Нистюк О.А.									
Утверд.	Нистюк О.А.									

- Name_Father – отчество, тип nvarchar;
- Sex – пол, тип nvarchar;
- Age – возраст, тип int;
- User_Type – идентификатор типа покупателя, тип int, внешний ключ.

Таблица STORE_ADMIN представляет данные об администраторе, состоит из столбцов:

- Admin_Id – идентификатор администратора магазина, тип int, первичный ключ;
- Admin_Login – логин администратора, тип nvarchar;
- Admin_Pass – пароль администратора, тип nvarchar;
- Admin_Type_Id – идентификатор типа администратора, тип int, внешний ключ;
- EMPL_Id – идентификатор сотрудника, тип int, внешний ключ.

Таблица PRODUCTS представляет собой товары магазина, состоит из столбцов:

- Prod_Num – идентификатор товара, тип int, первичный ключ;
- Prod_Id – штрихкод товара, тип nvarchar;
- Prod_Name – наименование товара, тип nvarchar;
- Prod_Type_Id – идентификатор типа товара, тип int, внешний ключ.
- Price_Purchase – цена закупки, тип int;
- Price_Sell – цена продажи, тип int;
- Weight – вес товара в граммах, тип int;
- Volume – объем товара в миллилитрах, тип int;

Стоит подробнее рассмотреть Prod_Id в таблице PRODUCTS, представляющий собой штрихкод товара – наносимая на поверхность маркировка, предоставляющая возможность считывания её техническими средствами, обозначающая номер товара. В торговле наиболее часто используется код EAN/UPC. Первоначально была разработана американская система UPC, содержащая в себе для кодировки товара 12 цифр. Однако весь диапазон кодов уже был занят для кодирования товаров США и Канады. Для товаров за пределами США был разработан EAN-13, обеспечивающий максимальную совместимость с кодировкой UPC. Решением стало добавление тринадцатой цифры в крайней слева позиции. После UPC стал подмножеством кодировки EAN-13 с первой цифрой 0. В связи с этим для Prod_Id были выбраны атрибуты nvarchar(13) check(LEN(Prod_Id) = 13) unique. Пример создания таблицы PRODUCTS, с описанным ранее штрихкодом, представлен ниже на рисунке 2.1.

```

Create Table PRODUCTS
(
    Prod_Num int identity(1,1) constraint PROD_PK PRIMARY KEY,
    Prod_Id nvarchar(13) check(LEN(Prod_Id) = 13) unique,
    Prod_Name nvarchar(50),
    Type_Prod_Id int constraint PRODUCT_TYPE_ID_FK foreign key references PRODUCT_TYPE(Prod_Type_Id),
    Price_Purchase numeric(6,2),
    Price_Sell numeric(6,2),
    Weight int,
    Volume int
);

```

Рисунок 2.1 – Пример создания таблицы PRODUCTS

Так же есть еще 7 таблиц для поддержания всей остальной логической структуры базы данных сети магазинов.

Таблица USER_TYPE представляет тип покупателей магазина, состоит из столбцов:

- Type_User_Id – идентификатор типа пользователя, тип int, первичный ключ;
- Type_User – наименование типа пользователя, тип nvarchar;
- Sale_Size – размер скидки пользователя, тип int.

Таблица TYPE_STORE представляет тип магазина, состоит из столбцов:

- Store_User_Id – идентификатор типа магазина, тип int, первичный ключ;
- Type – наименование типа магазина, тип nvarchar.

Таблица PRODUCT_TYPE представляет тип товаров в магазине, состоит из столбцов:

- Prod_Type_Id – идентификатор типа товара, тип int, первичный ключ;
- Prod_Type – наименование типа товара, тип nvarchar;
- Id_Dep – идентификатор отдела к которому относится тип товара, тип int, внешний ключ.

Таблица USER_BASKET представляет товары, купленные покупателем, состоит из столбцов:

- Id_Item – идентификатор покупки, тип int, первичный ключ;
- Id_User – идентификатор покупателя, тип int, внешний ключ;
- Product – идентификатор товара, тип int, внешний ключ;
- Date_Buy – дата покупки, тип date.

Таблица POSTS представляет должности сотрудников магазина, состоит из столбцов:

- Post_Id – идентификатор типа магазина, тип int, первичный ключ;
- Post – наименование типа магазина, тип nvarchar.

Таблица DEPARTMENTS представляет отделы магазина, состоит из столбцов:

- Dep_Id – идентификатор типа магазина, тип int, первичный ключ;
- Depart – наименование типа магазина, тип nvarchar.

Таблица TYPE_ADMIN представляет тип администратора, состоит из столбцов:

- Type_Id – идентификатор типа администратора, тип int, первичный ключ;

– Admin_Type– наименование типа администратора, тип nvarchar.

2.2 Хранимые процедуры

Хранимая процедура – объект базы данных, представляющий собой набор SQL-инструкций, который компилируется один раз и хранится на сервере.

При разработке курсового проекта было создано 40 процедур для следующих целей:

1. Выборка данных для таблиц;
2. Выборка данных для dropdown полей;
3. Выборка данных по поисковому запросу;
4. Заполнение таблиц 100 000 строк;
5. Вход администратора в аккаунт;
6. Удаление данных из таблиц;
7. Добавление данных в таблицы;
8. Изменение данных в таблицах;
9. Экспорт и импорт таблицы PRODUCTS в формате xml;
10. Резервное копирование и восстановление базы данных;

2.2.1 Выборка данных для таблиц

Для вывода данных из таблиц были написаны следующие процедуры: SEmpl, SProd, SStore, SUser, SProdOrder и SAdmin. Основной их особенностью является нумерование через row_number(), необходимый отсечение некоторого диапазона значений. Вывод таблицы размером в 100 000 строк на сайте занимал около 20 секунд, ограничение вывода до 100 строк и создание индексов позволило уменьшить время вывода таблицы до минимума.

Процедуры SProdOrder была создана с оглядкой на необходимость сортировки товаров. Сортировка возможна в обе стороны за счет ASC, по умолчанию, и DESC.

2.2.2 Выборка данных для dropdown полей

При добавлении новой строки или обновления уже существующей, необходимо, чтобы поля были валидными. Для упрощения валидации были созданы процедуры для выборки значения для dropdown полей, позволяя отменить ошибки валидирования с внешними ключами.

2.2.3 Выборка данных по поисковому запросу

Для вывода данных из таблиц по поисковому запросу были написаны следующие процедуры FindProd и OutBasket. FindProd выводит информацию о товаре из таблицы PRODUCTS при вводе штрихкода. OutBasket выводит список покупок при вводе идентификатора покупателя.

2.2.4 Заполнение таблиц 100 000 строк

Для заполнения таблиц было разработаны процедуры для генерации случайных строк и процедуры заполнения таблиц этими строками.

Процедуры заполнения таблиц: AI_USER, AI_STORE, AI_PROD, AI_BASKET и AI_EMPLOYEES. Они были разработаны соответственно для заполнения таблиц: USERS, STORES, PRODUCTS, USER_BASKET и EMPLOYEES.

Для генерации случайных имен была создана процедура CreateName, для генерации случайного штрихкода – CreateBarcode. Код процедур CreateName и CreateBarcode представлен в приложении Б.

2.2.5 Вход администратора в аккаунт

Процедура Login проверяет корректность логина и пароля, и, в случае корректно введенных пароля и логина, возвращает 1, в противном случае 0. Данная процедура представлена ниже на рисунке 2.2.

```

Create procedure Login
    @log nvarchar(13),
    @pass nvarchar(30)
AS
Begin
    Declare @password nvarchar(30);
    set @password = (select Admin_Pass from STORE_ADMIN where Admin_Login= @log);
    if @password = @pass
        select 1 as login;
    else
        select 0 as login;
End;

```

Рисунок 2.2 – Пример создания процедуры Login

2.2.6 Удаление данных из таблиц

Для управления базой данных были созданы процедуры удаления: SEmplDel, SProdDel, SStoreDel, SUserDel, SProdOrderDel, SAdminDel и OutBasketDel. Они были разработаны соответственно для удаления строк из таблиц: USERS, STORES, PRODUCTS, USER_BASKET, EMPLOYEES, STORE_ADMIN и USER_BASKET.

2.2.7 Добавление данных в таблицы

Для управления базой данных были созданы процедуры добавления новых строк: SEmplAdd, SProdAdd, SStoreAdd, SUserAdd, SProdOrderAdd, SAdminAdd и OutBasketAdd. Они были разработаны соответственно для заполнения таблиц: USERS, STORES, PRODUCTS, USER_BASKET, EMPLOYEES, STORE_ADMIN и USER_BASKET.

Процедура OutBasketAdd добавляет в строку Date_Buy таблицы USER_BASKET текущую дату с помощью GETDATE(), пример создания процедуры OutBasketAdd представлен ниже на рисунке 2.3.

```
Create Procedure OutBasketAdd
    @User int,
    @Id nvarchar(13)
AS
Begin
    Insert into USER_BASKET(Id_User, Product, Date_Buy)
    values(@User, (select Prod_Num from PRODUCTS where Prod_Id = @Id), CAST(GETDATE() AS DATE));
End;
```

Рисунок 2.3 – Пример создания процедуры OutBasketAdd

2.2.8 Изменение данных в таблицах

Для управления базой данных были созданы процедуры обновления: SEmplUpdate, SProdUpdate, SStoreUpdate, SUserUpdate, SProdOrderUpdate, SAdminUpdate. Они были разработаны соответственно для обновления данных в таблицах: USERS, STORES, PRODUCTS, USER_BASKET, STORE_ADMIN, EMPLOYEES, STORE_ADMIN.

2.2.9 Экспорт и импорт таблицы PRODUCTS в формате xml

Для экспорта таблицы PRODUCTS в формате xml была разработана процедура ExProdtoXml. Для формирования xml в select запросе используется конструкция FOR XML. Для вывода в файл была использована расширенная хранимая процедура xp_cmdshell. Процедура ExProdtoXml представлена ниже на рисунке 2.4. Отрывок из xml представлен в приложении В.

```
Create Procedure ExProdtoXml
AS
Begin
    SELECT Prod_Id, Prod_Name, Type_Prod_Id, Price_Purchase, Price_Sell, Weight, Volume from PRODUCTS
    FOR XML PATH('PRODUCT'), ROOT('PRODUCTS');

    --to use xp_cmdshell
    EXEC master.dbo.sp_configure 'show advanced options', 1
    RECONFIGURE WITH OVERRIDE
    EXEC master.dbo.sp_configure 'xp_cmdshell', 1
    RECONFIGURE WITH OVERRIDE;

    -- Save XML records to a file
    DECLARE @fileName nVARCHAR(100)
    DECLARE @sqlStr VARCHAR(1000)
    DECLARE @sqlCmd VARCHAR(1000)

    SET @fileName = 'D:\Subject\BD_COURSE\backups\STORES_CHAIN-Prod.xml'
    SET @sqlStr = 'USE STORES_CHAIN; SELECT Prod_Id, Prod_Name, Type_Prod_Id, Price_Purchase, Price_Sell, Weight,
        Volume from PRODUCTS FOR XML PATH(''PRODUCT''), ROOT(''PRODUCTS'') '
    SET @sqlCmd = 'bcp "' + @sqlStr + '" queryout ' + @fileName + ' -w -T'
    EXEC xp_cmdshell @sqlCmd;
End;
```

Рисунок 2.4 – Пример создания процедуры ExProdtoXml

Для импорта данных в таблицу PRODUCTS, из файла формата xml, была разработана процедура ImProdfromXml. Для получения xml файла и последующего

разбора со вставкой используется конструкция FROM OPENROWSET совместно с параметром BULK. Процедура ImProdfromXml представлена ниже на рисунке 2.5.

```

Create Procedure ImProdfromXml
AS
Begin
    INSERT INTO PRODUCTS (Prod_Id, Prod_Name, Type_Prod_Id, Price_Purchase, Price_Sell, Weight, Volume)
    SELECT
        MY_XML.PRODUCT.query('Prod_Id').value('.', 'NVARCHAR(13)'),
        MY_XML.PRODUCT.query('Prod_Name').value('.', 'VARCHAR(50)'),
        MY_XML.PRODUCT.query('Type_Prod_Id').value('.', 'INT'),
        MY_XML.PRODUCT.query('Price_Purchase').value('.', 'numeric(6,2)'),
        MY_XML.PRODUCT.query('Price_Sell').value('.', 'numeric(6,2)'),
        MY_XML.PRODUCT.query('Weight').value('.', 'INT'),
        MY_XML.PRODUCT.query('Volume').value('.', 'INT')

    FROM (SELECT CAST(MY_XML AS xml)
          FROM OPENROWSET(BULK 'D:\Subject\BD_COURSE\backups\Import.xml', SINGLE_BLOB) AS T(MY_XML)) AS T(MY_XML)
    CROSS APPLY MY_XML.nodes('PRODUCTS/PRODUCT') AS MY_XML (PRODUCT);
End;

```

Рисунок 2.5 – Пример создания процедуры ImProdfromXml

2.2.10 Резервное копирование и восстановление базы данных

Для копирования базы данных была разработана процедура ExDB. Для этого используется конструкция backup database, которая экспортирует базу данных в файл с расширением bak. Эта процедура представлена ниже на рисунке 2.6.

```

Create Procedure ExProdtoXml
AS
Begin
    SELECT Prod_Id, Prod_Name, Type_Prod_Id, Price_Purchase, Price_Sell, Weight, Volume from PRODUCTS
    FOR XML PATH('PRODUCT'), ROOT('PRODUCTS');

    --to use xp_cmdshell
    EXEC master.dbo.sp_configure 'show advanced options', 1
    RECONFIGURE WITH OVERRIDE
    EXEC master.dbo.sp_configure 'xp_cmdshell', 1
    RECONFIGURE WITH OVERRIDE;

    -- Save XML records to a file
    DECLARE @fileName nVARCHAR(100)
    DECLARE @sqlStr VARCHAR(1000)
    DECLARE @sqlCmd VARCHAR(1000)

    SET @fileName = 'D:\Subject\BD_COURSE\backups\STORES_CHAIN-Prod.xml'
    SET @sqlStr = 'USE STORES_CHAIN; SELECT Prod_Id, Prod_Name, Type_Prod_Id, Price_Purchase, Price_Sell, Weight,
        Volume from PRODUCTS FOR XML PATH(''PRODUCT''), ROOT(''PRODUCTS'') '
    SET @sqlCmd = 'bcp "' + @sqlStr + '" queryout ' + @fileName + ' -w -T'
    EXEC xp_cmdshell @sqlCmd;
End;

```

Рисунок 2.6 – Пример создания процедуры ExDB

Для восстановления базы данных была разработана процедура ImDB. Для этого используется конструкция restore database, которая экспортирует базу данных в файл с расширением bak. Эта процедура представлена ниже на рисунке 2.7.

```

use master;

--Restore DB
go
Create Procedure ImDB
AS
Begin
    ALTER DATABASE STORES_CHAIN
    SET SINGLE_USER
    WITH ROLLBACK IMMEDIATE;
    |
    RESTORE DATABASE STORES_CHAIN
        FROM disk = 'D:\Subject\BD_COURSE\backups\STORES_CHAIN.bak'
        WITH replace;
    ALTER DATABASE STORES_CHAIN
    SET MULTI_USER;
End;

```

Рисунок 2.7 – Пример создания процедуры ImDB

Для запуска и исполнения данной процедуры необходимо ее создавать в системной базе данных master. Это необходимо для избегания коллизий при восстановлении. Так же восстановление невозможно, при наличии подключенных к базе пользователей, поэтому переводим базу данных в монопольный режим с помощью set single_user и отменяем все изменения через параметр rollback immediate. После восстановления возвращаем базу данных в прежнее состояние через set multi_user.

2.3 Индексы

Индекс – объект базы данных, создаваемый с целью повышения производительности поиска данных. Таблицы в базе данных могут иметь большое количество строк, которые хранятся в произвольном порядке, и их поиск по заданному критерию путём последовательного просмотра таблицы строка за строкой может занимать много времени. В связи с необходимостью выборки таблиц, заполненных минимум на 100 000 строк, было принято решение разработать 24 индекса для таблиц: PRODUCTS, EMPLOYEES, USERS, USER_BASKET, STORE_ADMIN и STORES.

3 Обоснование технических приемов программирования

При проектировании программного средства в качестве программной платформы для сервера был выбран Node.js с пакетами Express.js, fs, body-parser, msnodesqlv8 и node-mssql. Непосредственно для взаимодействия с Microsoft SQL Server нам необходимы только пакеты msnodesqlv8 и node-mssql. Схема взаимодействия с Microsoft SQL Server представлена на рисунке 4.1.

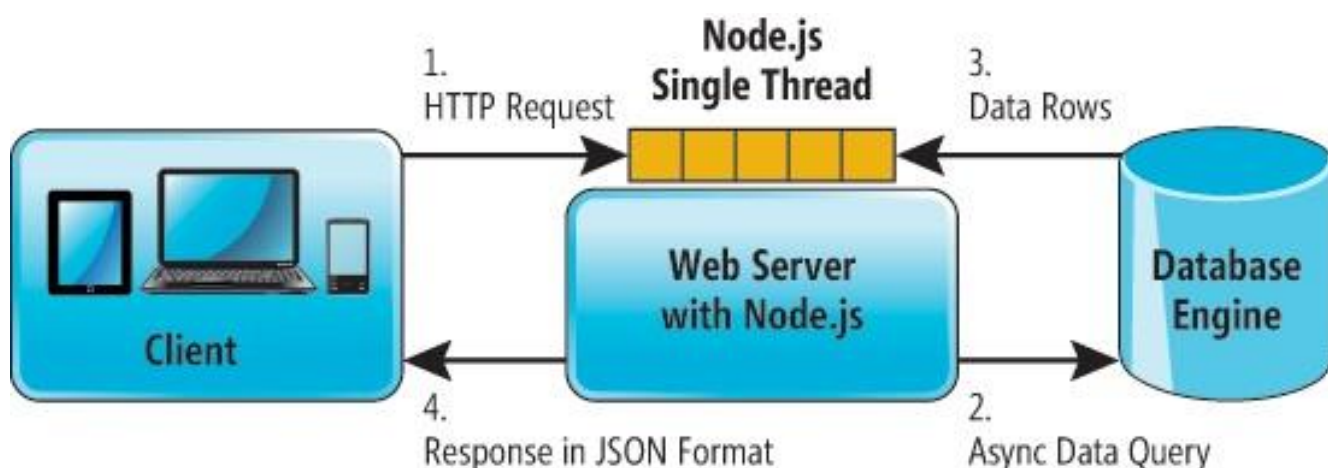


Рисунок 3.1 – Схема взаимодействия с Microsoft SQL Server

Для работы с базой данных используется пакет node-mssql, как клиент базы данных. TDS драйвером выступает msnodesqlv8, позволяя упростить работу с СУБД. Пример конфигурации, необходимой для подключения к Microsoft SQL Server, представлена ниже на рисунке 3.2.

```

let connectionPool;
const config =
{
  "driver": "msnodesqlv8",
  "connectionString": "Driver={SQL Server Native Client 11.0};Server={DESKTOP-U4BLHC6};Database={STORES_CHAIN};Trusted_Connection={yes};"
};

```

Рисунок 3.2 – Конфигурация для подключения к Microsoft SQL Server

Далее необходимо получить доступ к пулу подключений, задействовав ранее представленную конфигурацию. Пример подключения представлен ниже на рисунке 3.3.

				КП 03.00.ПЗ			
	ФИО	Подпись	Дата				
Разраб.	Матюх А.А.			3 Обоснование технических приёмов программирования		Лит.	Лист
Провер.	Нистюк О.А.						Листов
						1	2
Н. контр.	Нистюк О.А.					БГТУ 71171075, 2019	
Утверд.	Нистюк О.А.						

```

constructor()
{
    connectionPool = new sql.ConnectionPool(config).connect().then(pool =>
    {
        console.log('Connected to MSSQL')
        return pool
    }).catch(err => console.log('Connection Failed: ', err));
}

```

Рисунок 3.3 – Подключение к Microsoft SQL Server

После запроса к базе данных, представленном ниже на рисунке 3.4, данные помещаются в пул и передаются серверу.

```

Get(tab, end, start, order)
{
    console.log(`Exec ${tab} ${order} ${end}, ${start}`);
    return connectionPool.then(pool => pool.query(`Exec ${tab} ${order} ${end}, ${start}`));
}

```

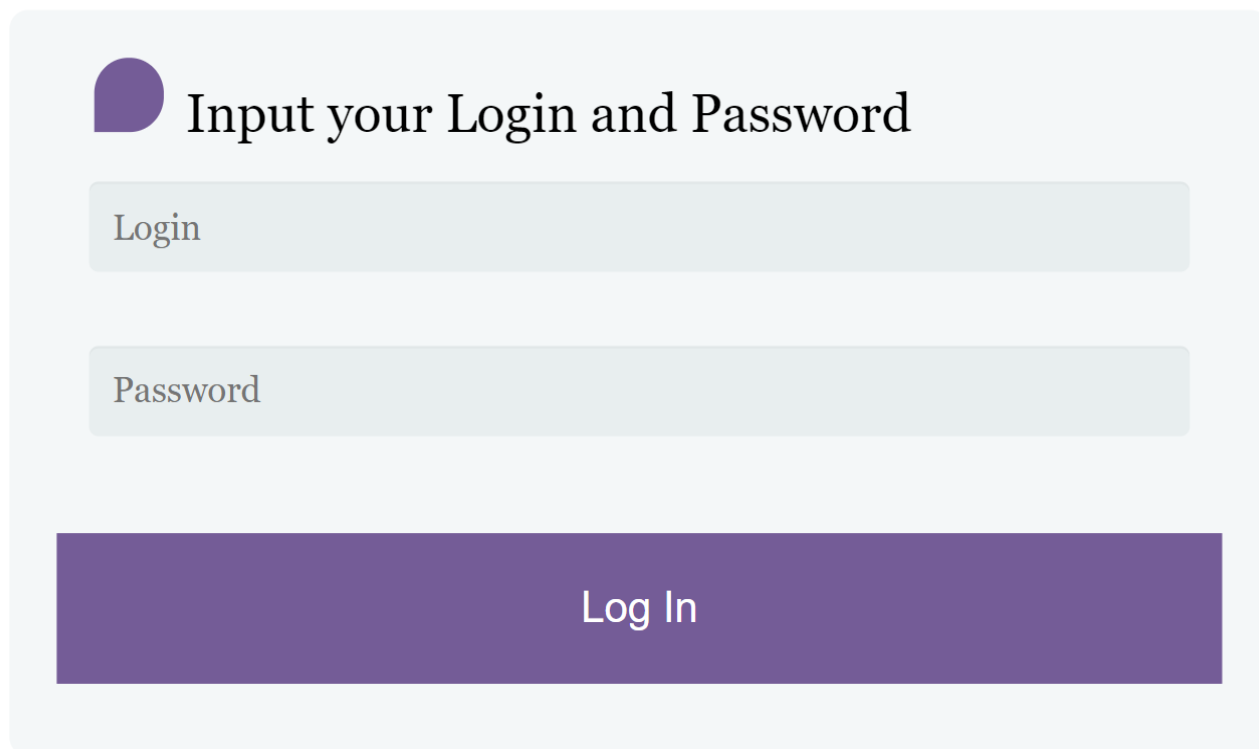
Рисунок 3.4 – Запрос к Microsoft SQL Server

Получив данные, сервер их обрабатывает и, запарсив результат в формат JSON, отправляет их клиенту.

4 Тестирование, проверка работоспособности и анализ данных

Протестируем для начала страницу авторизации.

Введём логин и пароль. Страница авторизации продемонстрировано ниже на рисунке 4.1.

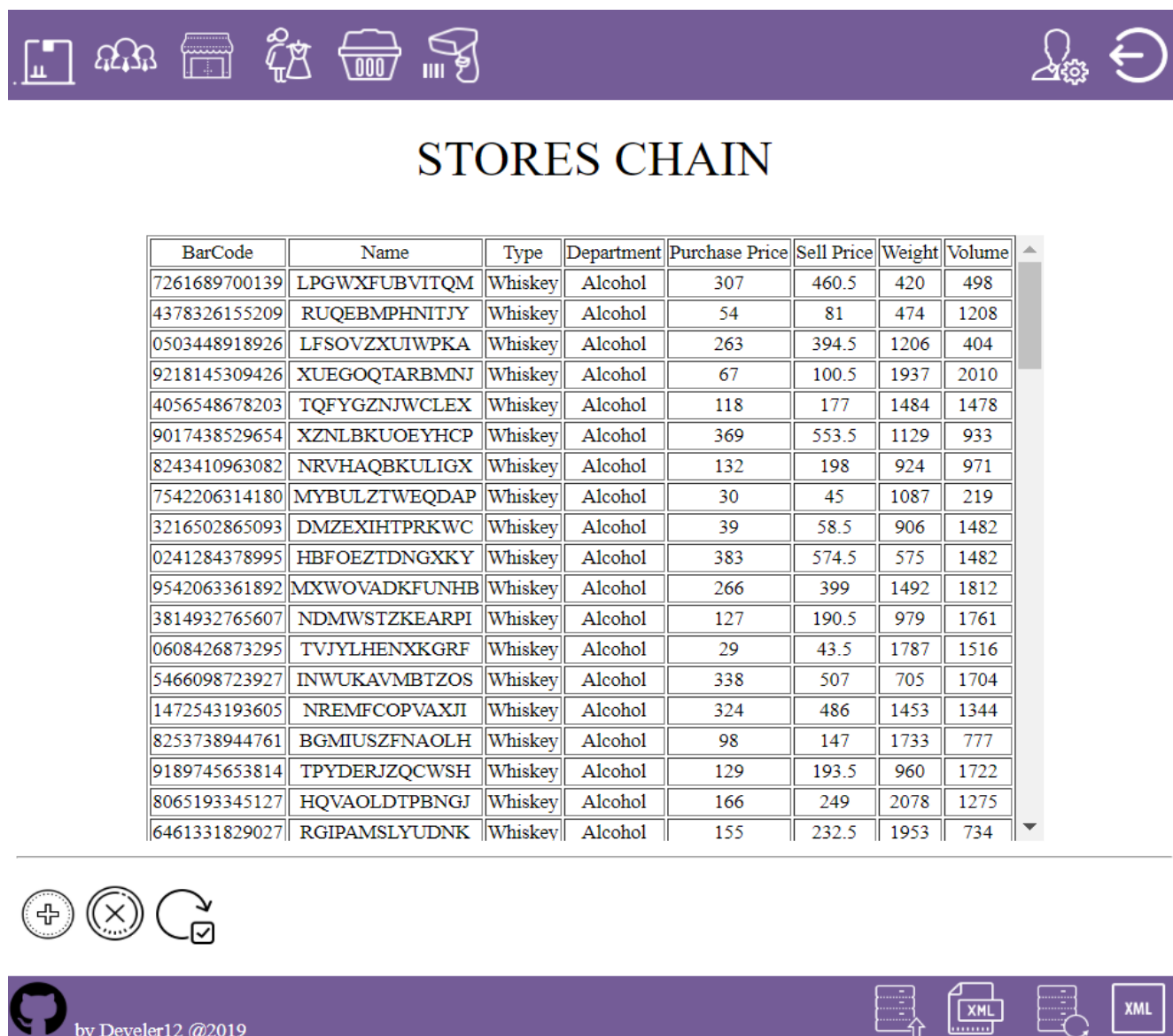


The image shows a login interface with a light blue background. At the top left is a purple circular logo. To its right is the text 'Input your Login and Password'. Below this are two light blue rectangular input fields. The first field is labeled 'Login' and the second is labeled 'Password'. At the bottom of the form is a large purple rectangular button with the text 'Log In' in white.

Рисунок 4.1 – Страница авторизации

В случае не верно введенных данных, система вернет пользователя снова на эту же страницу. Если же данные верны, то пользователь перейдет на страницу «Stores Chain». На которой расположены элементы управления. Эта страница представлена на рисунке 4.2.

				КП 04.00.ПЗ						
	ФИО	Подпись	Дата							
Разраб.	Матюх А.А.			4 Тестирование, проверка работоспособности и анализ данных			Лит.	Лист	Листов	
Провер.	Нистюк О.А.							1	6	
							БГТУ 71171075, 2019			
Н. контр.	Нистюк О.А.									
Утверд.	Нистюк О.А.									



BarCode	Name	Type	Department	Purchase Price	Sell Price	Weight	Volume
7261689700139	LPGWXFUBVITQM	Whiskey	Alcohol	307	460.5	420	498
4378326155209	RUQEBMPHNITJY	Whiskey	Alcohol	54	81	474	1208
0503448918926	LFSOVZXUTWPKA	Whiskey	Alcohol	263	394.5	1206	404
9218145309426	XUEGOQTARBMNJ	Whiskey	Alcohol	67	100.5	1937	2010
4056548678203	TQFYGZJWCLEX	Whiskey	Alcohol	118	177	1484	1478
9017438529654	XZNLBKUOEYHCP	Whiskey	Alcohol	369	553.5	1129	933
8243410963082	NRVHAQBKULIGX	Whiskey	Alcohol	132	198	924	971
7542206314180	MYBULZTWEQDAP	Whiskey	Alcohol	30	45	1087	219
3216502865093	DMZEXIHTPRKWC	Whiskey	Alcohol	39	58.5	906	1482
0241284378995	HBFOEZTDNGXKY	Whiskey	Alcohol	383	574.5	575	1482
9542063361892	MXWOVADKFUNHB	Whiskey	Alcohol	266	399	1492	1812
3814932765607	NDMWSTZKEARPI	Whiskey	Alcohol	127	190.5	979	1761
0608426873295	TVJYLHENXKGRF	Whiskey	Alcohol	29	43.5	1787	1516
5466098723927	INWUKAVMBTZOS	Whiskey	Alcohol	338	507	705	1704
1472543193605	NREMFCOPVAXJI	Whiskey	Alcohol	324	486	1453	1344
8253738944761	BGMIUSZFNAOLH	Whiskey	Alcohol	98	147	1733	777
9189745653814	TPYDERJZQCWSH	Whiskey	Alcohol	129	193.5	960	1722
8065193345127	HQVAOLDTPBNGJ	Whiskey	Alcohol	166	249	2078	1275
6461331829027	RGIPAMSLYUDNK	Whiskey	Alcohol	155	232.5	1953	734

Рисунок 4.2 – Страница «Stores Chain»

На верхней панели управления страницы «Stores Chain» слева направо расположены кнопки для:

1. Просмотра, добавления, обновления и удаления товаров;
2. Просмотра, добавления, обновления и удаления работниками;
3. Просмотра, добавления, обновления и удаления магазинами;
4. Просмотра, добавления, обновления и удаления покупателями;
5. Просмотра, добавления и удаления покупок покупателей;
6. Поиска товара по штрихкоду;
7. Просмотра, добавления, обновления и удаления администраторов;
8. Выхода из аккаунта администратора.

На нижней панели управления страницы «Stores Chain» слева направо расположены кнопки для:

1. Перехода на страницу курсового проекта на Github;
2. Резервного восстановления базы данных;
3. Импорта данных из xml;

4. Резервного копирования базы данных;
5. Экспорта данных в xml
6. Выхода из аккаунта администратора.

Выведем список товаров. Пример списка товаров приведен на рисунке 4.3.

STORES CHAIN

BarCode	Name	Type	Department	Purchase Price	Sell Price	Weight	Volume
0694715188743	694715188743	Interior	For Home	2	2	2	2
3428573009921	ABCFQHSELOWTG	Foto and Video	Electronic	441	661.5	1473	1199
9718276925084	ABCHODLGVQEPY	Tinctures	Alcohol	512	768	839	1260
6357181295736	ABCHTGSWDLKPE	Hand and foot care	Cosmetics	352	528	1465	792
9072190855763	ABCKDRNEHPOSX	Pasta	Grocery	282	423	608	1100
6052339970281	ABCNHPXQGUYDV	Nuts, dried fruits	Grocery	300	450	2066	1062
7024113483956	ABCQKPYXUMVFR	Products for child	For Office	409	613.5	174	208
0285198137234	ABCRPZLXJMSYU	Hygiene	Cosmetics	509	763.5	628	1671
1574886223931	ABDKIWNJPMFUQ	Chewing gum	Sweets	40	60	1812	628
7869105045612	ABDUIONPRVLMK	Butter	Milk	586	879	1108	143
9943560582717	ABDVRYTIOQGEH	Computer	Electronic	375	562.5	1816	971
9514736768382	ABEHDNZUYOLQJ	Eggs	Meet	181	271.5	1469	103
8097667153984	ABENZPUKRQVIS	Other alcohol	Alcohol	40	60	188	1354
0701391274245	ABEOZITGHXUVL	Audio	Electronic	276	414	462	158
9366082014418	ABESCXNVOZPTU	Berries	Fruit Vegetable	584	876	561	1636
1005833952697	ABESWDZVPJICM	Porridge	Child Food	84	126	969	528
0694715188746	ABFDWNTORLQVP	Canned meat	Child Food	140	210	1969	1173
2471053968548	ABFRHSMQXIWKJ	Pasta	Grocery	435	652.5	679	873
7269825416079	ABFVXOCJMZTIH	Fruit and vegetable purees	Child Food	541	811.5	2086	1803
6243483179015	ABGKXTPRYLHNS	Berries	Fruit Vegetable	276	414	1210	229



Рисунок 4.3 – Вывод списка товаров

Ниже списка есть кнопки добавления, обновления и удаления товара. Попробуем добавить новую товарную позицию. Добавление товара представлено на рисунке 4.4.

6243483179015	ABGKXTPRYLHNS	Berries	Fruit Vegetable	276	414	1210	229
---------------	---------------	---------	-----------------	-----	-----	------	-----

Рисунок 4.4 – Добавление товарной позиции

В случае ошибки, например при вводе не валидного значения, такого, как уже существующий штрихкод, будет выведена соответственное сообщение. Пример сообщения об ошибке представлен на рисунке 4.5.



Рисунок 4.5 – Сообщения об ошибке

Попробуем обновить информацию в только что добавленной товарной позиции. Обновление товара представлено на рисунке 4.6.

7269825416079	ABFVXOCJMZTIH	Fruit and vegetable purees	Child Food	541	811.5	2086	1803
6243483179015	ABGKXTPRYLHNS	Berries	Fruit Vegetable	276	414	1210	229

0694715188741

ProductWrode

Mushrooms

Fruit|Vegetable

Purchase Price

Sell Price

Weight

Volume

Input

Рисунок 4.6 – Обновление товарной позиции

Попробуем удалить товарную позицию. Удаление товара представлено на рисунке 4.7.

7269825416079	ABFVXOCJMZTIH	Fruit and vegetable purees	Child Food	541	811.5	2086	1803
6243483179015	ABGKXTPRYLHNS	Berries	Fruit Vegetable	276	414	1210	229

0694715188741

Delete

Рисунок 4.7 – Удаление товарной позиции

В списке товаров реализована сортировка по категориям, для этого нужно нажать на название столбца. Если необходима сортировка в другом порядке, то следует нажать еще раз.

Далее выведем список работников магазина. Пример списка работников приведен на рисунке 4.8.

Id	First Name	Second Name	Father Name	Post	Sex	Age	Experience	Store	Country	Department
100003	23	das	dsfg	Cleaner	F	23	34	MQWFPOLZR	Poland	Drinks
75492	ABCGZSYNRH	RSNWKUHQZD	PAGKEYYHVC	Adviser	M	58	16	MQWFPOLZR	Poland	Milk
24173	ABCHZQMGTJ	UNVDPGXSKJ	ZFUMWSLIOV	Manager	M	59	11	SMWOGBFLNQ	Germany	Alcohol
24211	ABCMZJUFQE	AGYVUOLBMJ	GXWAPIBLYC	Manager	M	56	37	UEHWKMBVTL	Germany	Electronic
66972	ABCXOLRTPQ	ORICMKAWZU	KPYZGHBFEF	Manager	M	39	0	RZBKXYWAEQ	Belarus	Child Food
84454	ABDCKUGLHS	YBSEIOXUDK	ZDOXICNRKE	Cashier	M	64	16	VGJBSNRAOM	Russia	Meet
80383	ABDJWRKXVF	JQLRSTCDYM	UYGFTKCBIM	Assistanse	F	56	9	YJNMBVCSPW	Poland	Cosmetics
84596	ABDQZYPTHV	RZOPDQMUTY	WPNAOSTZMJ	Assistanse	F	55	22	DXZWFLVTJA	France	Sweets
20914	ABDRHNYUJW	LJOKTCWDFG	EQBISZTHMA	President	F	46	23	RVJTXZBIQW	Germany	Sweets
20426	ABDUFJGIVK	KRCAIEOPSV	DRJSZQPFHV	Manager	F	24	0	WQFYONRJZX	France	For Children
87851	ABEGQVRIDX	ONPFRHEYJ	GWXDJNTQSY	Manager	F	41	21	HWAMOFCDUJ	France	Milk
63846	ABELIRGUQH	FWNPMVKCID	ZEYFWMTHGO	Adviser	M	64	19	MDVEZRNJBH	France	For Home
19458	ABEMVHUFWK	BIUOFDVLHT	ULNEGPAUTO	Administrator	F	28	5	MWNYTHZRV	Poland	For Garden
77278	ABESQVKWYC	LDJAFWQTKX	FLAQUMSWHJ	President	F	58	16	ESPRMJUQBY	Poland	Grocery
92692	ABFQJCURGZ	TDVZQFIHSM	XJUOPTBMQH	DB Admin	M	64	35	BTPHYRASKM	France	Grocery
31582	ABFRXLNEI	RFAQZMYDOW	HZMYCIBSJW	Cleaner	M	60	12	SZJYEGAPMB	Belarus	Electronic
90546	ABFXDTSQWU	GVDCSUNQWE	NMFLCGOPVY	Vice president	F	64	1	BNACJTZHUR	Poland	Grocery
65205	ABGEPWQCXO	NXSGPEBYDI	HUEYZTGMRD	Assistanse	M	41	5	WFGROQLVZS	Belarus	Bakery
54621	ABGOIHSDMF	XNHBWQLRAP	QPLVTCESBM	Vice president	F	34	7	ULCPWYQERF	Russia	Sweets
657	ABGRUHTLZY	AGZSBCTYIQ	YXIBKGPLTS	Administrator	F	24	2	JWDIOAUEGB	Belarus	Stock

Рисунок 4.8 – Вывод списка работников

Выведем список магазинов. Пример списка магазинов приведен на рисунке 4.9.

Id	Store	Country	Town	Address	Size	Type	Workers
99999	UVNIZTCRJG	France	SNYHWLVPIA	EQGDTCLWF	2630	GroserMarket	2
99998	UKLRMPVTZD	France	NZVEUDYABG	CAQLPMZNXB	2871	MiniMarket	1
99997	HWOXIZUCPY	Germany	GPSHXZTWLM	FXBHNWPADO	797	HyperMarket	2
99996	HEMPFCKVYJ	France	UDHRTSVBCZ	RZLCXETOPK	2895	GroserMarket	1
99993	PIWBJXLMCV	France	XWTQFMLJSA	IBZWPOYGFD	498	MiniMarket	1
99992	NJXCGEKYFS	Russia	LYDGXFVHPT	SKCFWRXMAT	2453	HyperMarket	3
99990	QWGCVNTKPI	France	QACXNGLDIY	INECZAHPUK	2929	MiniMarket	1
99989	VQYGHKJPDE	Belarus	JBUDEQWXS	YDKIOVTBSW	451	SupreMarket	1
99988	YNQRCHIFGP	Poland	CEUWQDAJTL	DQOKNLSHUA	2882	Department	3
99985	GIZCPWTD OE	Russia	BHNCYQMJRT	XMRLICBDUV	1591	Department	1
99982	BOWMXYQPAR	Poland	FPAKUCYBNW	TNQUIYPGLE	2921	HyperMarket	2
99981	SVUPLKWBYI	Poland	JYBKFXQPWS	CTYDLREIPM	1241	HyperMarket	1
99980	ASBETHRNPY	Belarus	DFLRXOBCUE	SIMPNGCUQW	616	HyperMarket	1
99979	GPIYBUELMT	Germany	DMABUWVLRJ	XDJAFVYPIL	1648	Discount	1
99978	ZNBIERPACQ	Russia	QSAUCNMVZT	ASODZTMHOC	2202	HyperMarket	2

Рисунок 4.9 – Вывод списка магазинов

Попробуем вывести список покупателей. Пример списка покупателей приведен на рисунке 4.10.

Id	First Name	Second Name	Father Name	Sex	Age	Type	Sale
100004	23	23	dsfg	F	99	Student_Coll	0.01
100000	NYLJGKRXEP	RUCKIHGLSD	FJKOVLNXDY	M	37	Pensioner	0.03
99997	KSUCQNPEYW	OZNQEYABVT	UHYLDCQIZO	M	12	Student_Coll	0.01
99996	AUYLGEOHRZ	ULYRBOXSCF	YSKQPIWAJH	F	68	Student_Univ	0.03
99995	WFXLVPOCBJ	BMUJFSRYNC	XZFJYTBUCW	F	46	Student_Coll	0.01
99994	WAGQLDKVOH	DCLTBUEGON	NVZADWREOI	F	44	Student_Univ	0.03
99993	CPEHYGFISB	EKDNRLLOUMF	VQKLNDYOXF	M	24	Student_Coll	0.01
99992	LMOWGDKTNU	CWHGOTZKDX	HFGLZORIDP	F	10	Student_Coll	0.01
99991	EQRWTXCMNH	OPMXTNVIRL	OFYPBLGDMZ	M	32	Invalid	0.03
99990	NWPRSOJTGU	OUVPMWTENJ	YLKEMFNHAB	F	64	Student_Schl	0.02

Рисунок 4.10 – Вывод списка покупателей

Далее выведем список покупок, для этого сначала необходимо ввести идентификатор покупателя. Пример списка покупок приведен на рисунке 4.11.

12

BarCode	Date Buy	Name	Type	Department	Purchase Price	Sell Price	Weight	Volume
2178260435094	2019-12-17	NSEBOILMAXUQG	Soils, fertilizers, pesticides	For Garden	405	607.5	239	1039
2178260435094	2019-12-17	NSEBOILMAXUQG	Soils, fertilizers, pesticides	For Garden	405	607.5	239	1039
2178260435094	2019-12-15	NSEBOILMAXUQG	Soils, fertilizers, pesticides	For Garden	405	607.5	239	1039
2178260435094	2019-12-15	NSEBOILMAXUQG	Soils, fertilizers, pesticides	For Garden	405	607.5	239	1039
2178260435094	2019-12-15	NSEBOILMAXUQG	Soils, fertilizers, pesticides	For Garden	405	607.5	239	1039
2178260435094	2019-12-15	NSEBOILMAXUQG	Soils, fertilizers, pesticides	For Garden	405	607.5	239	1039
9526818492670	2019-12-13	ITGQORCXYWLEZ	Vegetables	Fruit Vegetable	30	45	846	1950
2178260435094	2019-12-11	NSEBOILMAXUQG	Soils, fertilizers, pesticides	For Garden	405	607.5	239	1039

Рисунок 4.11 – Вывод списка покупок

Найдем товар по штрих коду, для этого сначала необходимо ввести штрихкод необходимого товара. Поиска товара приведен на рисунке 4.12.

2178260435094

BarCode	Name	Type	Department	Purchase Price	Sell Price	Weight	Volume
2178260435094	ProductWrode	Berries	Fruit Vegetable	140	200.99	7	8

Рисунок 4.12 – Поиск товара по штрихкоду

Попробуем вывести список администраторов. Пример списка администраторов приведен на рисунке 4.13.

Admin ID	Login	Type	Employee ID
15	cenia	Admin_St	4
1	Admin	Main	1

Рисунок 4.13 – Вывод списка администраторов

А теперь подробнее рассмотрим кнопки в нижнем углу, приведенные на рисунке 4.14.

toys	For Children	120	107	570	071
Audio	Electronic	534	801	715	1940
Nuts, dried fruits	Grocery	132	198	1230	557
Coffee	Drinks	579	868.5	1025	485
Face care	Cosmetics	6	9	1899	1489
TV	Electronic	463	694.5	1625	484
Foto and Video	Electronic	196	294	1530	1198
Flour	Grocery	558	837	575	477
Ice Cream	Milk	358	537	1228	253
For construction	Electronic	449	673.5	946	1651
Hygiene	Cosmetics	392	588	301	760



Рисунок 4.14 – Кнопки в нижнем углу

Начнем с крайней правой кнопки, она отвечает за экспорт таблицы PRODUCTS в файл формата xml, который будет сохранен на стороне сервера. Левее расположена кнопка резервного копирования базы данных, которая будет сохранена на стороне сервера в формате bak. Еще левее расположена кнопка импорта данных из xml файла, расположенного на стороне сервера, в таблицу PRODUCTS. Еще левее расположена кнопка резервного восстановления базы данных. На ней остановимся по подробнее.

Кнопка резервного восстановления базы данных запускает данный процесс на стороне СУБД и автоматически выходит из аккаунта администратора, возвращая его на страницу авторизации. Так как базе данных для восстановления необходим режим монополии, в связи с чем сервер подключается к системной базе данных

master, а СУБД переходит в режим монополии, пользователь не должен пытаться авторизоваться в течении некоторого времени, иначе может потребоваться перезапуск сервера, так как возникнет критическая ошибка, связанная с отказом подключения к Microsoft SQL Server.

Заключение

В процессе решения поставленной задачи была достигнута поставленная цель по созданию базы данных «Сеть супермаркетов», которая в купе с приложением формирует сайт для управления сетью магазинов. Основой целью курсового проекта стало проектирование базы данных для дальнейшей интеграции с приложением, которое помогло облегчить взаимодействие с базой данных посредством программного интерфейса. При разработке выполнены следующие пункты:

- Регистрация и авторизация администраторов магазина;
- Разделение приложение на сервер и клиент;
- Технология резервного копирования и восстановления базы данных;
- Сортировка товара;
- Просмотр информации о сотрудниках;
- Просмотр информации о товарах;
- Просмотр информации о супермаркетах;
- Поиск товара по штриховому полю;
- Импорт и экспорт XML.

В курсовом проекте были реализованы следующие задачи:

- Создание базы данных;
- Создание сервера;
- Создание клиента с пользовательским интерфейсом;
- Реализация функций работы приложения;
- Тестирование программного продукта.

Данный проект сложно сравнить с аналогами, ведь для каждой сети магазинов системы разрабатываются индивидуально. По сути, данный проект является показательным примером базы данных для различных сетей магазинов, позволяя охватить необходимый минимум требований для их администрирования.

В соответствии с полученным результатом работы программы можно сделать вывод, что разработанная программа работает верно, а требования технического задания выполнены в полном объеме.

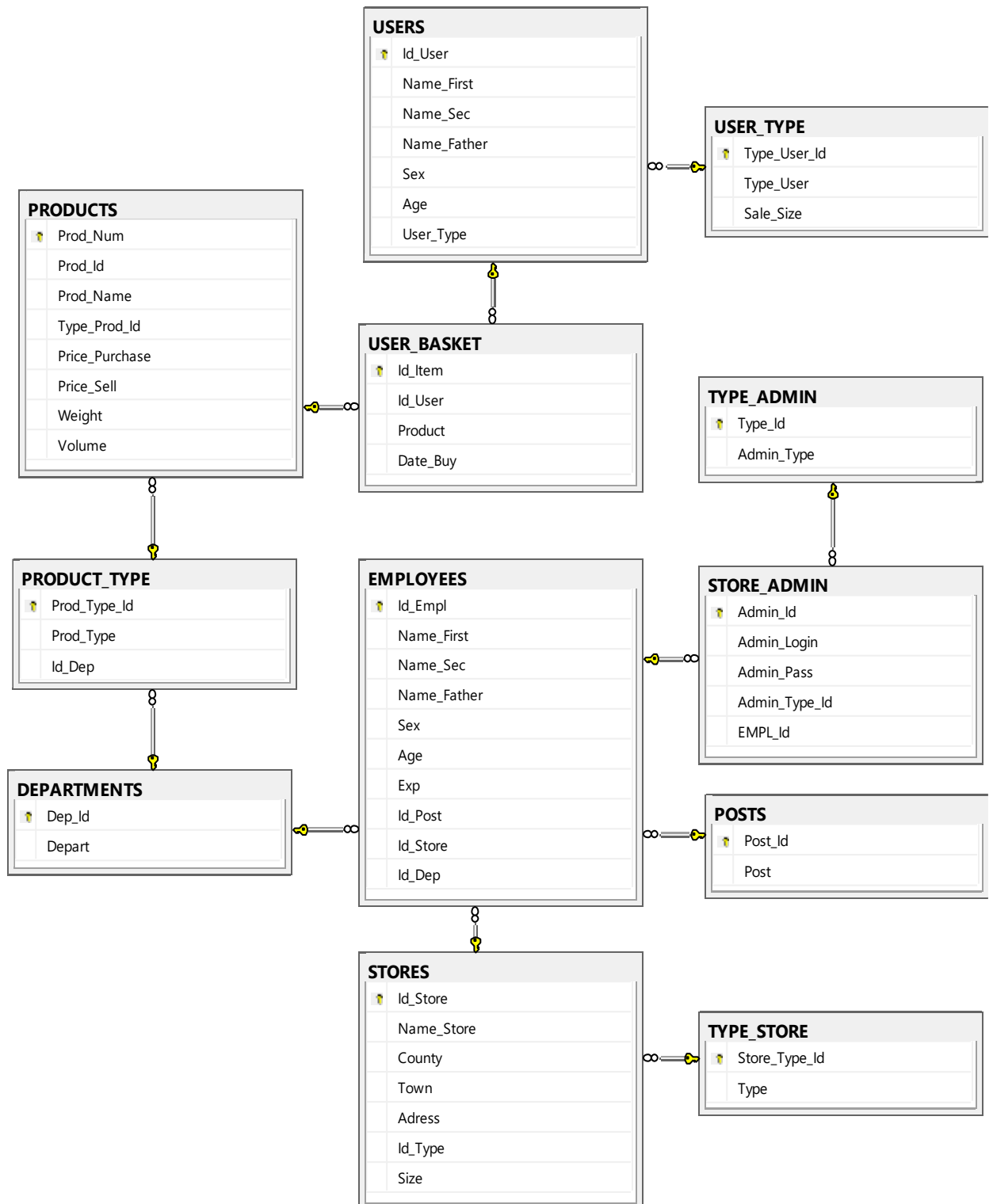
				КП 00.00.ПЗ			
	ФИО	Подпись	Дата				
Разраб.	Матюх А.А.			Заключение	Лит.	Лист	Листов
Провер.	Нистюк О.А.					1	1
					БГТУ 71171075, 2019		
Н. контр.	Нистюк О.А.						
Утверд.	Нистюк О.А.						

Список литературы

1. Node.js v11.15.0 Documentation [Электронный ресурс] / OpenJS Foundation. – Режим доступа: <https://nodejs.org/docs/latest-v11.x/api>. – Дата доступа: 28.10.2019.
2. Express 4.x API [Электронный ресурс] / StrongLoop, Inc. – Режим доступа: <https://expressjs.com/en/4x/api.html>. – Дата доступа: 28.10.2019.
3. Mssql [Электронный ресурс] / OpenJS Foundation. – Режим доступа: <https://www.npmjs.com/package/mssql>. – Дата доступа: 28.10.2019.
4. RESTORE Statements (Transact-SQL) [Электронный ресурс] / Microsoft. – Режим доступа: <https://docs.microsoft.com/en-us/sql/t-sql/statements/restore-statements-transact-sql?view=sql-server-ver15>. – Дата доступа: 04.11.2019.
4. BACKUP (Transact-SQL) [Электронный ресурс] / Microsoft. – Режим доступа: <https://docs.microsoft.com/en-us/sql/t-sql/statements/backup-transact-sql?view=sql-server-ver15>. – Дата доступа: 04.11.2019.
6. Restore a Database Backup Using SSMS [Электронный ресурс] / Microsoft. – Режим доступа: <https://docs.microsoft.com/en-us/sql/relational-databases/backup-restore/restore-a-database-backup-using-ssms?view=sql-server-ver15>. – Дата доступа: 15.11.2019.

				КП 00.00.ПЗ						
	ФИО	Подпись	Дата							
Разраб.	Матюх А.А.			Список литературы			Лит.	Лист	Листов	
Провер.	Нистюк О.А.								1	1
							БГТУ 71171075, 2019			
Н. контр.	Нистюк О.А.									
Утверд.	Нистюк О.А.									

Приложение А



Приложение Б

```

go
-----Random name alghorythm-----
--COMPLETED!!
Create Procedure CreateName
    @size integer,
    @Name char(50) OUTPUT
AS
Begin
    SET @Name = (
    SELECT
        c1 AS [text()]
    FROM
        (
            SELECT TOP (@size) c1
            FROM
                (
                    VALUES
                        ('A'), ('B'), ('C'), ('D'), ('E'), ('F'), ('G'), ('H'), ('I'), ('J'),
                        ('K'), ('L'), ('M'), ('N'), ('O'), ('P'), ('Q'), ('R'), ('S'), ('T'),
                        ('U'), ('V'), ('W'), ('X'), ('Y'), ('Z')
                ) AS T1(c1)
            ORDER BY ABS(CHECKSUM(NEWID()))
        ) AS T2
    FOR XML PATH('')
    );
End;

go
-----Random BarCode alghorythm-----
--COMPLETED!!
Create Procedure CreateBarcode
    @BarC char(13) OUTPUT
AS
Begin
    SET @BarC = (
    SELECT
        c1 AS [text()]
    FROM
        (
            SELECT TOP (13) c1
            FROM (VALUES ('0'), ('1'), ('2'), ('3'), ('4'), ('5'), ('6'), ('7'), ('8'), ('9'),
                        ('0'), ('1'), ('2'), ('3'), ('4'), ('5'), ('6'), ('7'), ('8'), ('9')) AS T1(c1)
            ORDER BY ABS(CHECKSUM(NEWID()))
        ) AS T2
    FOR XML PATH('')
    );
End;

```


Приложение В

<PRODUCTS>

<PRODUCT>

<Prod_Id>0843251726618</Prod_Id>

<Prod_Name>MOHNICDQSBTYK</Prod_Name>

<Type_Prod_Id>75</Type_Prod_Id>

<Price_Purchase>337.00</Price_Purchase>

<Price_Sell>505.50</Price_Sell>

<Weight>377</Weight>

<Volume>1559</Volume>

</PRODUCT>

<PRODUCT>

<Prod_Id>8859342031427</Prod_Id>

<Prod_Name>OHFPGDYKSMNWV</Prod_Name>

<Type_Prod_Id>19</Type_Prod_Id>

<Price_Purchase>568.00</Price_Purchase>

<Price_Sell>852.00</Price_Sell>

<Weight>496</Weight>

<Volume>2012</Volume>

</PRODUCT>