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:**

«Перечень заказов»

Выполнил студент Матюх Арсений Артурович

(Ф.И.О.)

Руководитель проекта пр.-стаж. Бурмакова А.В.

(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. Бурмакова А.В.

(учен. степень, звание, должность, Ф.И.О., подпись)

Нормоконтролер: пр.-стаж. Бурмакова А.В.

(учен. степень, звание, должность, Ф.И.О., подпись)

Курсовой проект защищен с оценкой

Минск 2020

Реферат

ФИО

Подпись

Дата

Лист

1

КП 00.00.ПЗ

Разраб.

Матюх А.А.

Провер.

Бурмакова А.В.

Н. контр.

Бурмакова А.В.

Утверд.

Бурмакова А.В.

Реферат

Лит.

Листов

1

БГТУ 71171075, 2019

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

NODE.JS, JAVASCRIPT, MICROSOFT SQL SERVER, EXPRESS.JS, TEDIOUS, SEQUELIZE, MVC

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.

Abstract

ФИО

Подпись

Дата

Лист

1

КП 00.00.ПЗ

Разраб.

Матюх А.А.

Провер.

Бурмакова А.В.

Н. контр.

Бурмакова А.В.

Утверд.

Бурмакова А.В.

Abstract

Лит.

Листов

1

БГТУ 71171075, 2019

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.

Содержание

ФИО

Подпись

Дата

Лист

1

КП 00.00.ПЗ

Разраб.

Матюх А.А.

Провер.

Бурмакова А.В.

Н. контр.

Бурмакова А.В.

Утверд.

Бурмакова А.В.

Содержание

Лит.

Листов

1

БГТУ 71171075, 2019

[Введение 7](#_Toc27472552)

[1 Аналитический обзор литературы 8](#_Toc27472553)

[2 Разработка программной системы 11](#_Toc27472554)

[2.1 Разработка модели базы данных 11](#_Toc27472555)

[2.2 Хранимые процедуры 14](#_Toc27472556)

[2.2.1 Выборка данных для таблиц 14](#_Toc27472557)

[2.2.2 Выборка данных для dropdown полей 14](#_Toc27472558)

[2.2.3 Выборка данных по поисковому запросу 14](#_Toc27472559)

[2.2.4 Заполнение таблиц 100 000 строк 15](#_Toc27472560)

[2.2.5 Вход администратора в аккаунт 15](#_Toc27472561)

[2.2.6 Удаление данных из таблиц 15](#_Toc27472562)

[2.2.7 Добавление данных в таблицы 15](#_Toc27472563)

[2.2.8 Изменение данных в таблицах 16](#_Toc27472564)

[2.2.9 Экспорт и импорт таблицы PRODUCTS в формате xml 16](#_Toc27472565)

[2.2.10 Резервное копирование и восстановление базы данных 17](#_Toc27472566)

[2.3 Индексы 18](#_Toc27472567)

[3 Обоснование технических приемов программирования 19](#_Toc27472568)

[4 Тестирование, проверка работоспособности и анализ данных 21](#_Toc27472569)

[Заключение 29](#_Toc27472570)

[Список литературы 30](#_Toc27472571)

[Приложение А 31](#_Toc27472572)

[Приложение Б 32](#_Toc27472573)

[Приложение В 33](#_Toc27472574)

Введение

ФИО

Подпись

Дата

Лист

1

КП 00.00.ПЗ

Разраб.

Матюх А.А.

Провер.

Нистюк О.А.

Н. контр.

Нистюк О.А.

Утверд.

Нистюк О.А.

Введение

Лит.

Листов

1

БГТУ 71171075, 2019

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 был 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 следующий пакеты:

* Express.js – предоставляет собой слой фундаментальных функций для упрощения разработки веб-приложений;
* Body-parser – пакет для парсинга body;
* Msnodesqlv8 – 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.

Аналитический обзор литературы

ФИО

Подпись

Дата

Лист

1

КП 01.00.ПЗ

Разраб.

Матюх А.А.

Провер.

Бурмакова А.В.

Н. контр.

Бурмакова А.В.

Утверд.

Бурмакова А.В.

1 Аналитический обзор литературы

Лит.

Листов

3

БГТУ 71171075, 2019

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.

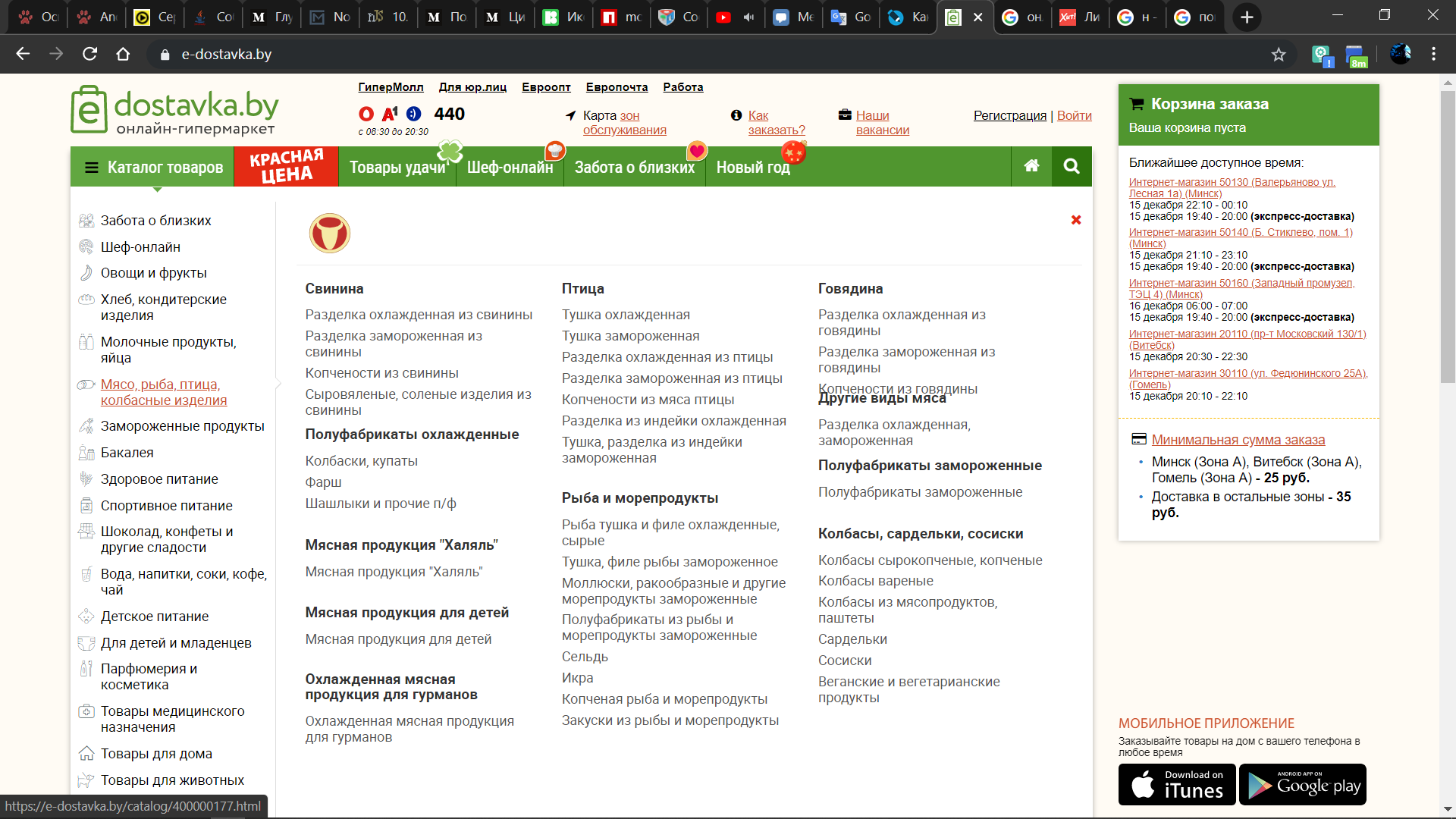


Рисунок 1.1 – Пример каталога «e-dostavka.by»

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.

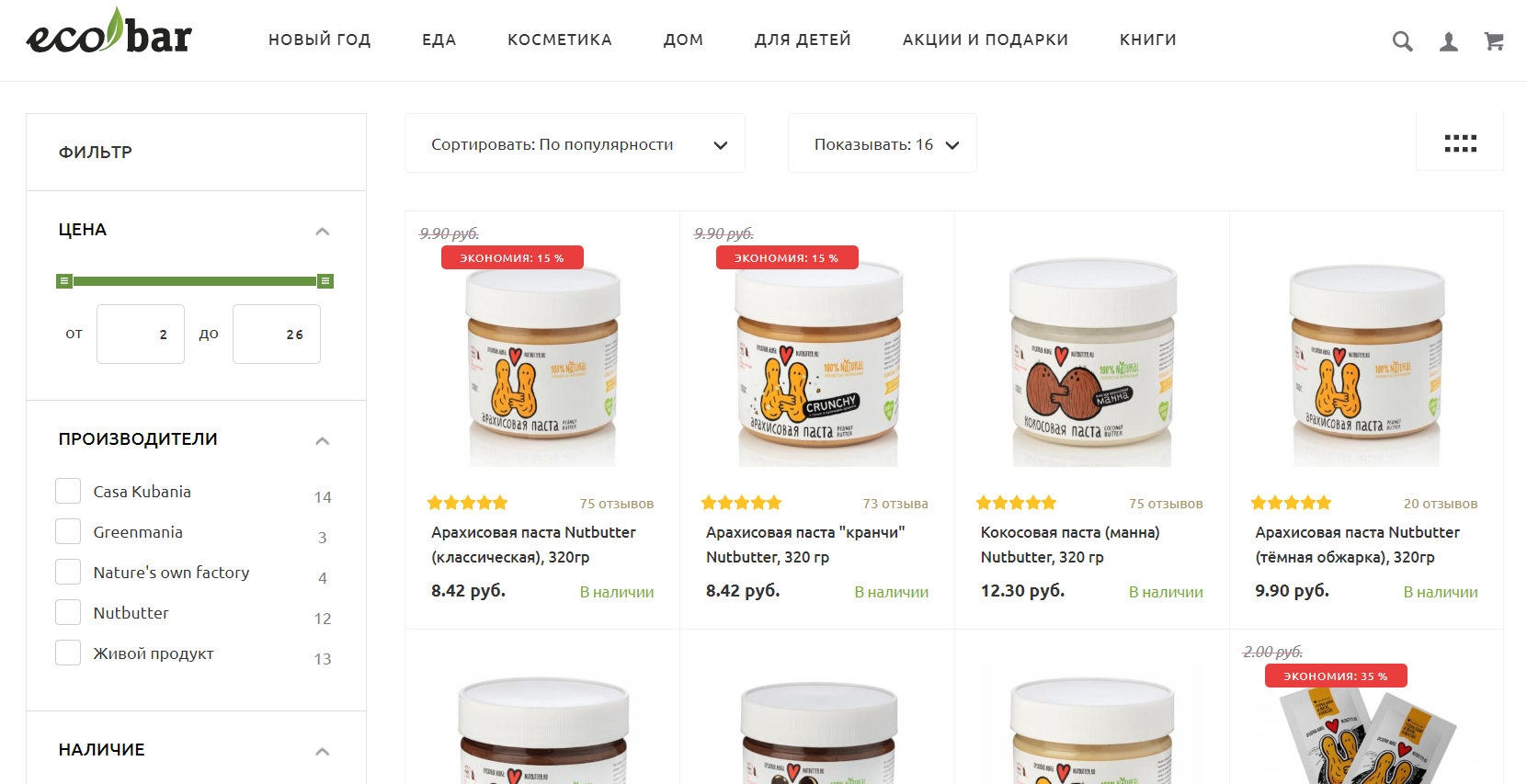


Рисунок 1.2 – Пример сайта «ecobar.by»

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.

Разработка программной системы

ФИО

Подпись

Дата

Лист

1

КП 02.00.ПЗ

Разраб.

Матюх А.А.

Провер.

Нистюк О.А.

Н. контр.

Бурмакова А.В.

Утверд.

Бурмакова А.В.

2 Разработка программной системы

Лит.

Листов

8

БГТУ 71171075, 2019

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.

ТаблицаSTORESпредставляет 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;
* 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 был выбраны атрибуты nvarchar(13) check(LEN(Prod\_Id) = 13) unique. Пример создания таблицы PRODUCTS, с описанным ранее штрихкодом, представлен ниже на рисунке 2.1.

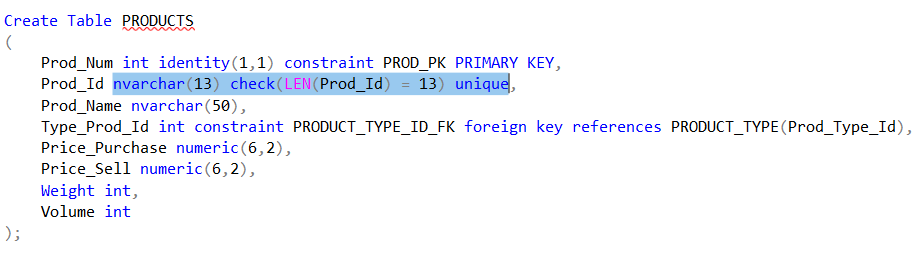


Рисунок 2.1 – Пример создания таблицы PRODUCTS

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\_STOREпредставляет тип магазина, состоит из столбцов:

* Store\_User\_Id – идентификатор типа магазина, тип int, первичный ключ;
* Type – наименование типа магазина, тип nvarchar.

Таблица PRODUCT\_TYPEпредставляет 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 DEPARTMENTSпредставляет отделы магазина, состоит из столбцов:

* Dep\_Id – идентификатор типа магазина, тип int, первичный ключ;
* Depart – наименование типа магазина, тип nvarchar.

Таблица TYPE\_ADMINпредставляет 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 отмести 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.

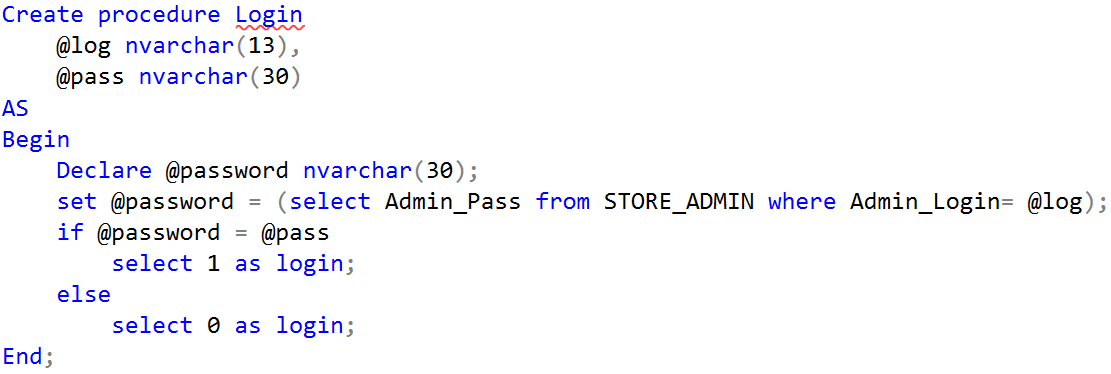


Рисунок 2.2 – Пример создания процедуры Login

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.

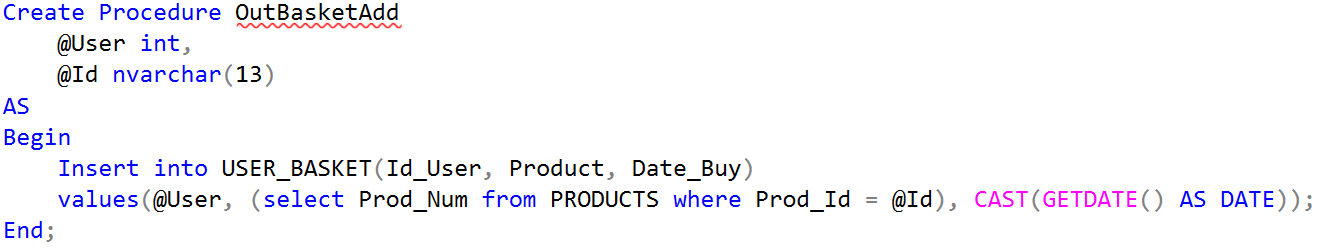


Рисунок 2.3 – Пример создания процедуры OutBasketAdd

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.

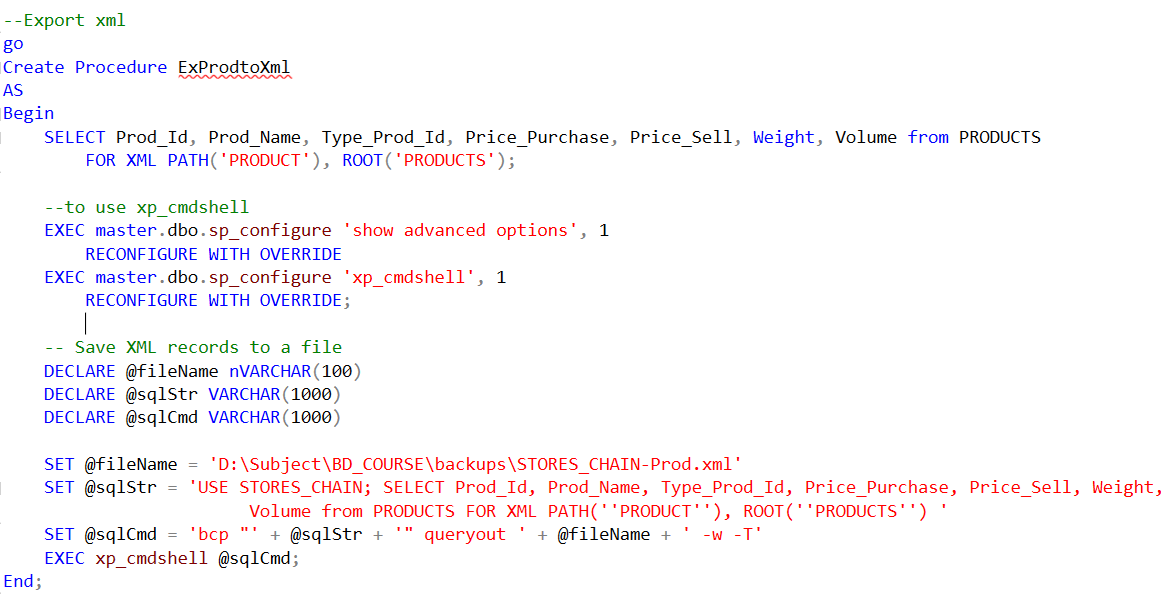


Рисунок 2.4 – Пример создания процедуры ExProdtoXml

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.

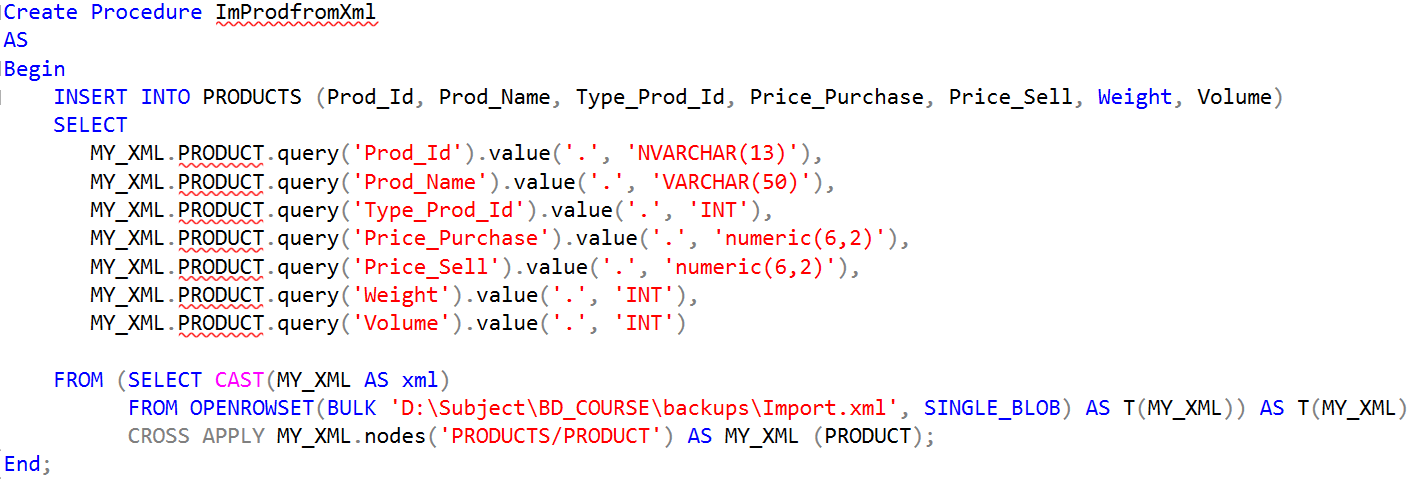


Рисунок 2.5 – Пример создания процедуры ImProdfromXml

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.

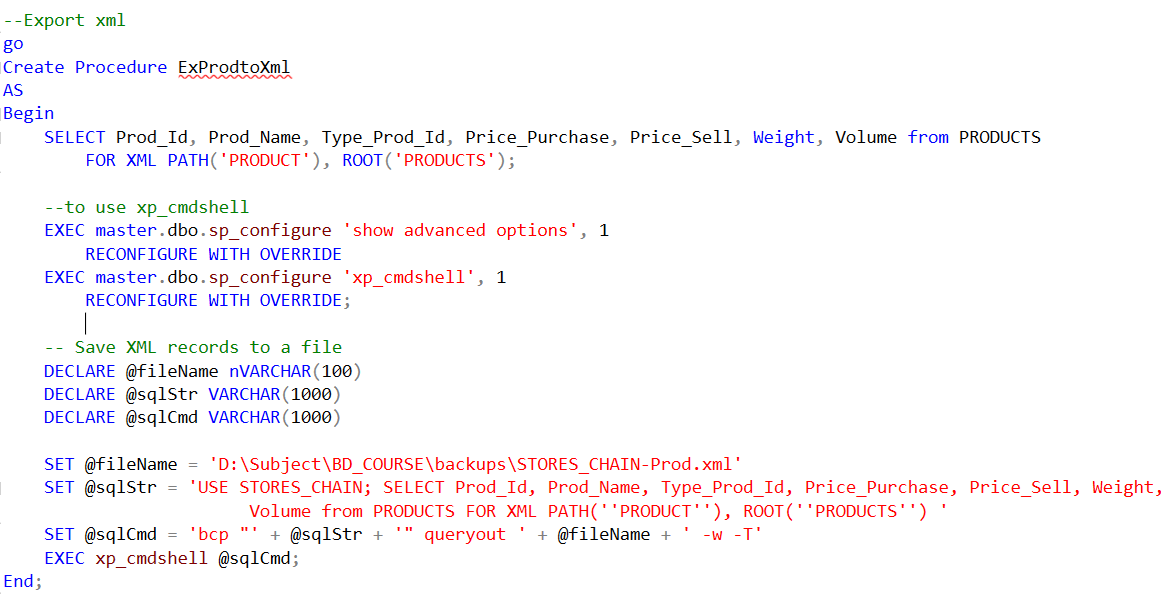


Рисунок 2.6 – Пример создания процедуры ExDB

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.

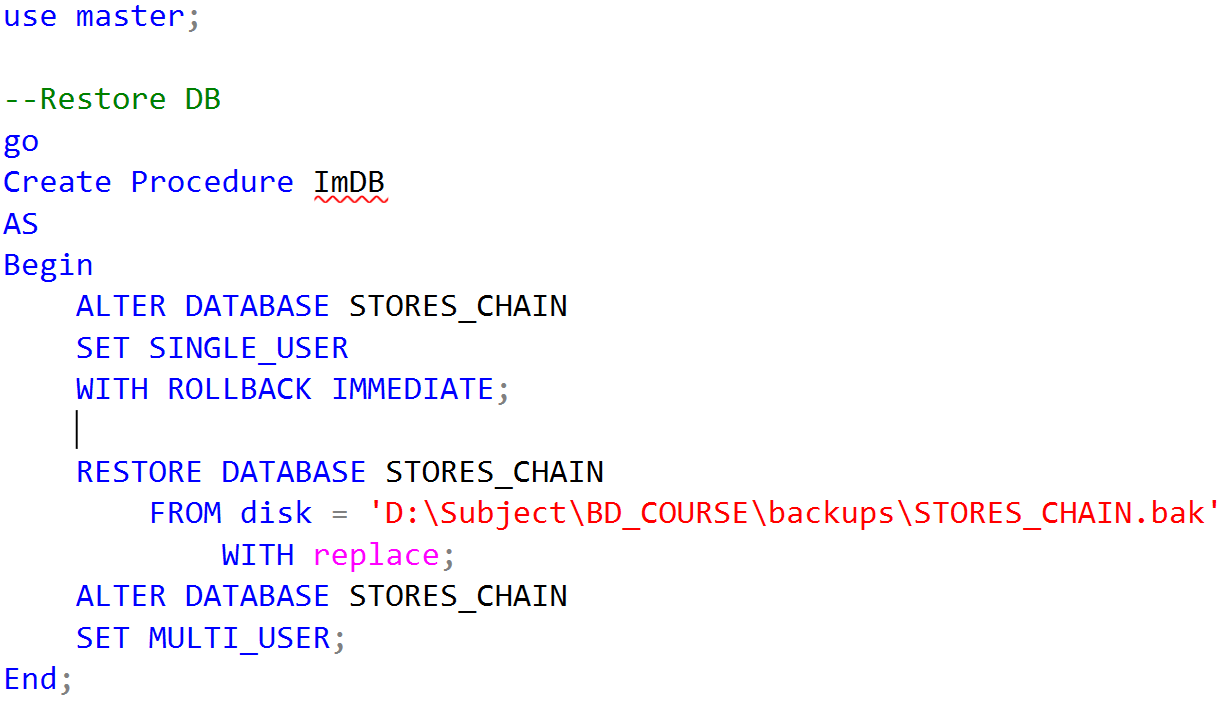


Рисунок 2.7 – Пример создания процедуры ImDB

Для запуска и исполнения данной процедуры необходимо ее создавать 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.

Обоснование технических приемов программирования

ФИО

Подпись

Дата

Лист

1

КП 03.00.ПЗ

Разраб.

Матюх А.А.

Провер.

Бурмакова А.В.

Н. контр.

Бурмакова А.В.

Утверд.

Бурмакова А.В.

3 Обоснование технических приёмов программирования

Лит.

Листов

2

БГТУ 71171075, 2019

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.

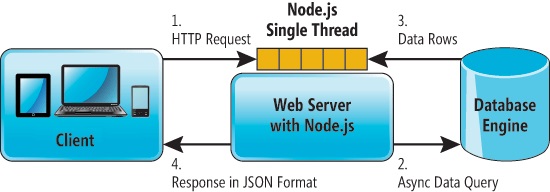


Рисунок 3.1 – Схема взаимодействия с Microsoft SQL Server

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.

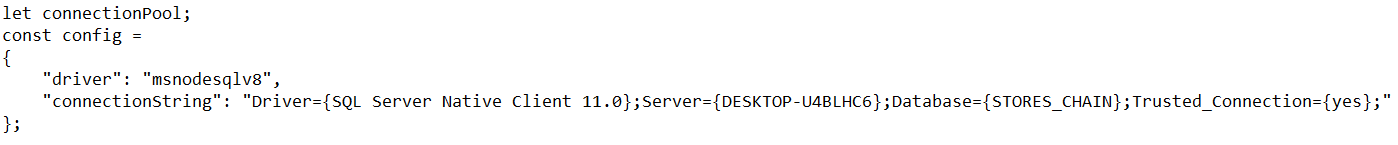


Рисунок 3.2 – Конфигурация для подключения к Microsoft SQL Server

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

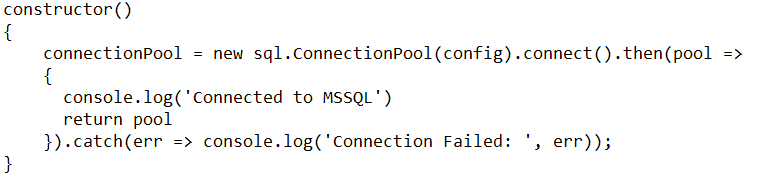


Рисунок 3.3 – Подключение к Microsoft SQL Server

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

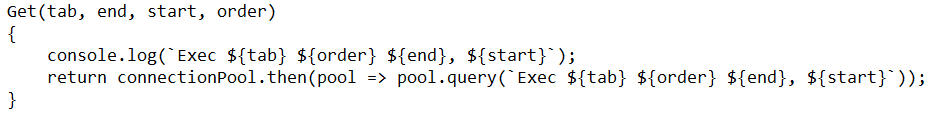


Рисунок 3.4 – Запрос к Microsoft SQL Server

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

Тестирование, проверка работоспособности и анализ данных

ФИО

Подпись

Дата

Лист

1

КП 04.00.ПЗ

Разраб.

Матюх А.А.

Провер.

Бурмакова А.В.

Н. контр.

Бурмакова А.В.

Утверд.

Бурмакова А.В.

4 Тестирование, проверка работоспособности и анализ данных

Лит.

Листов

6

БГТУ 71171075, 2019

Протестируем для начала страницу авторизации.

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

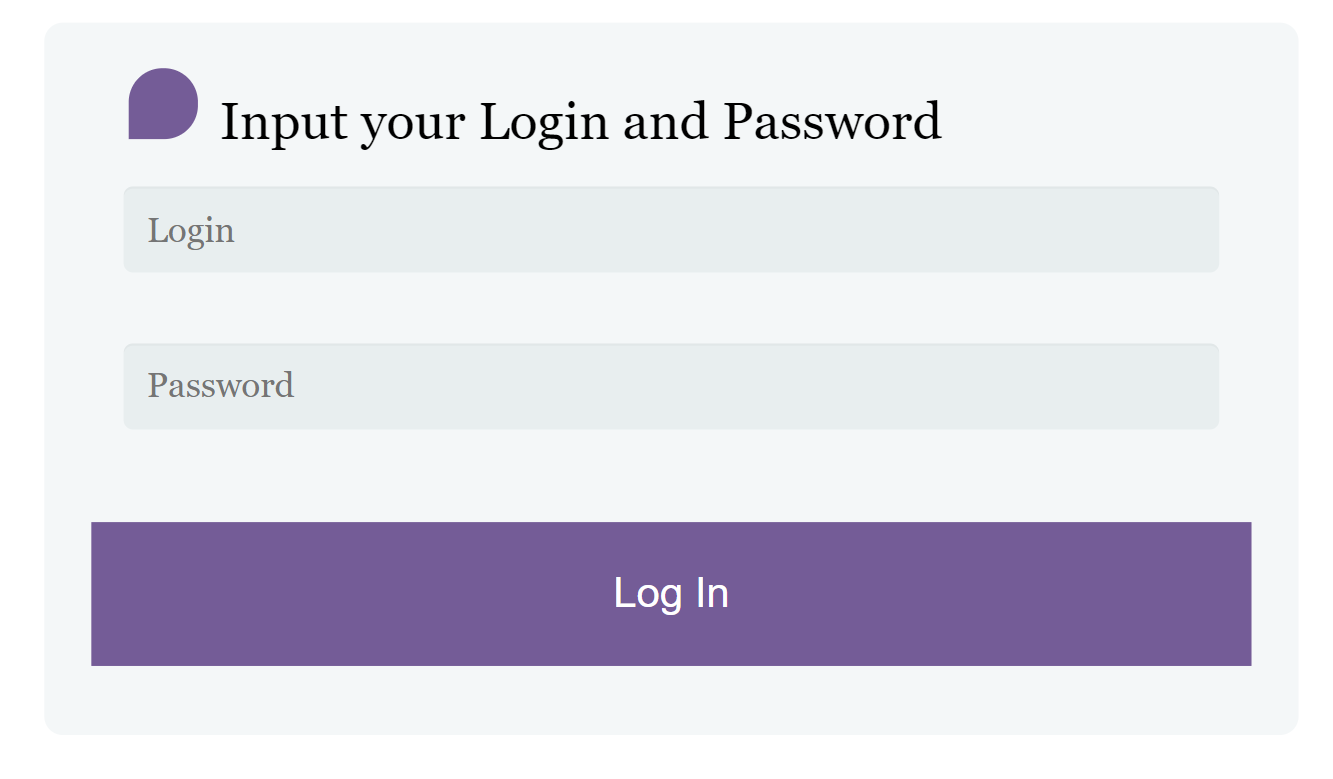


Рисунок 4.1 – Страница авторизации

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.

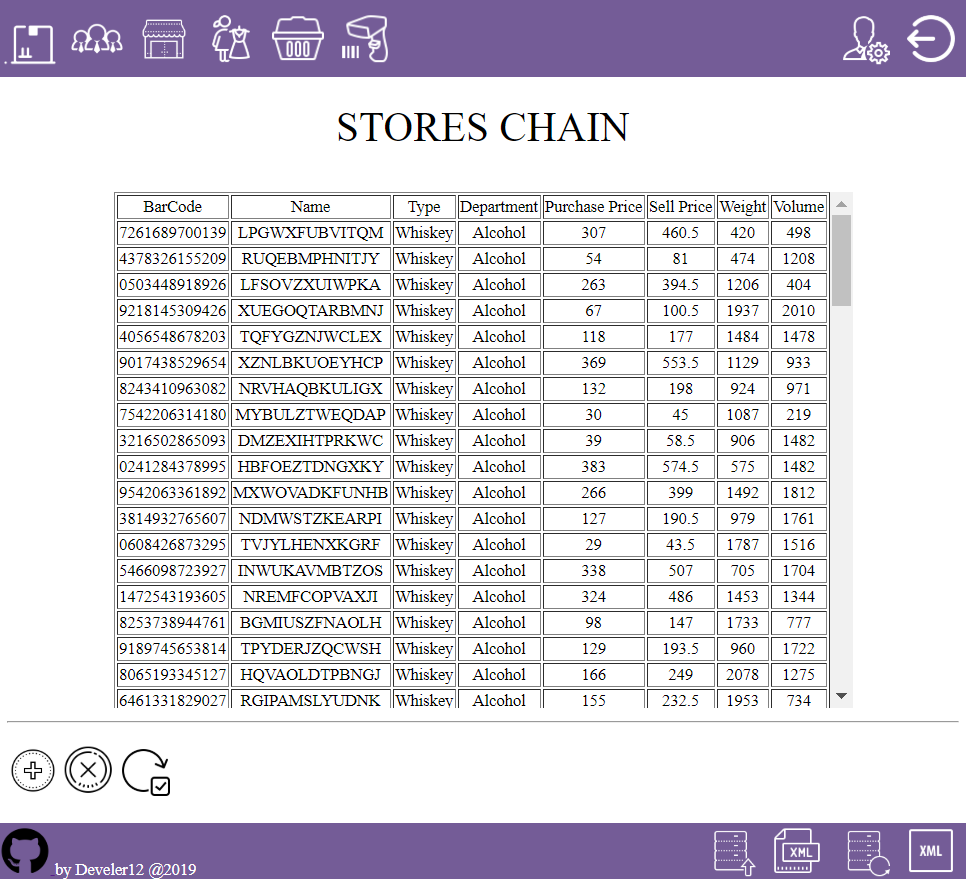


Рисунок 4.2 – Страница «Stores Chain»

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.

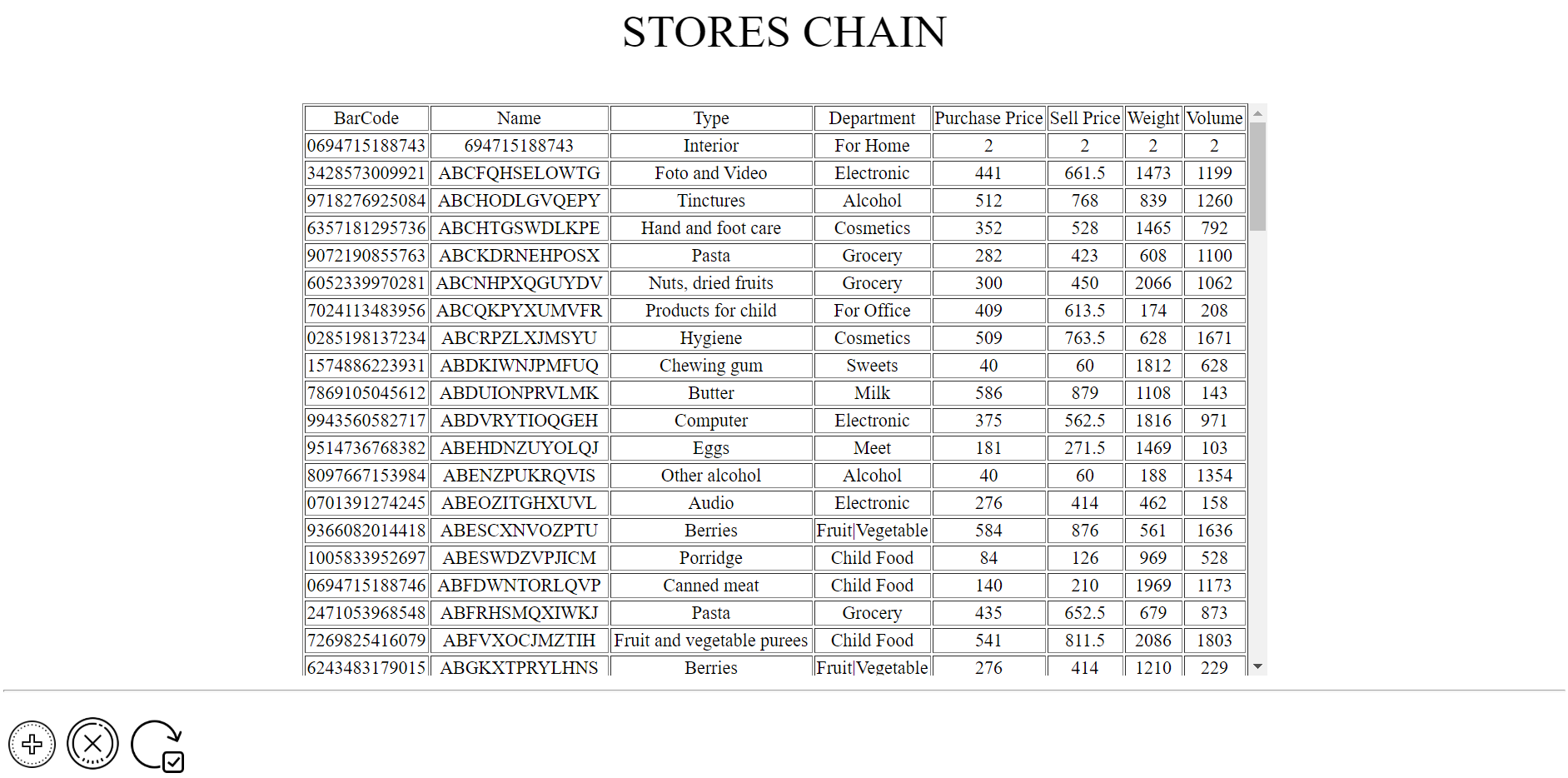


Рисунок 4.3 – Вывод списка товаров

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

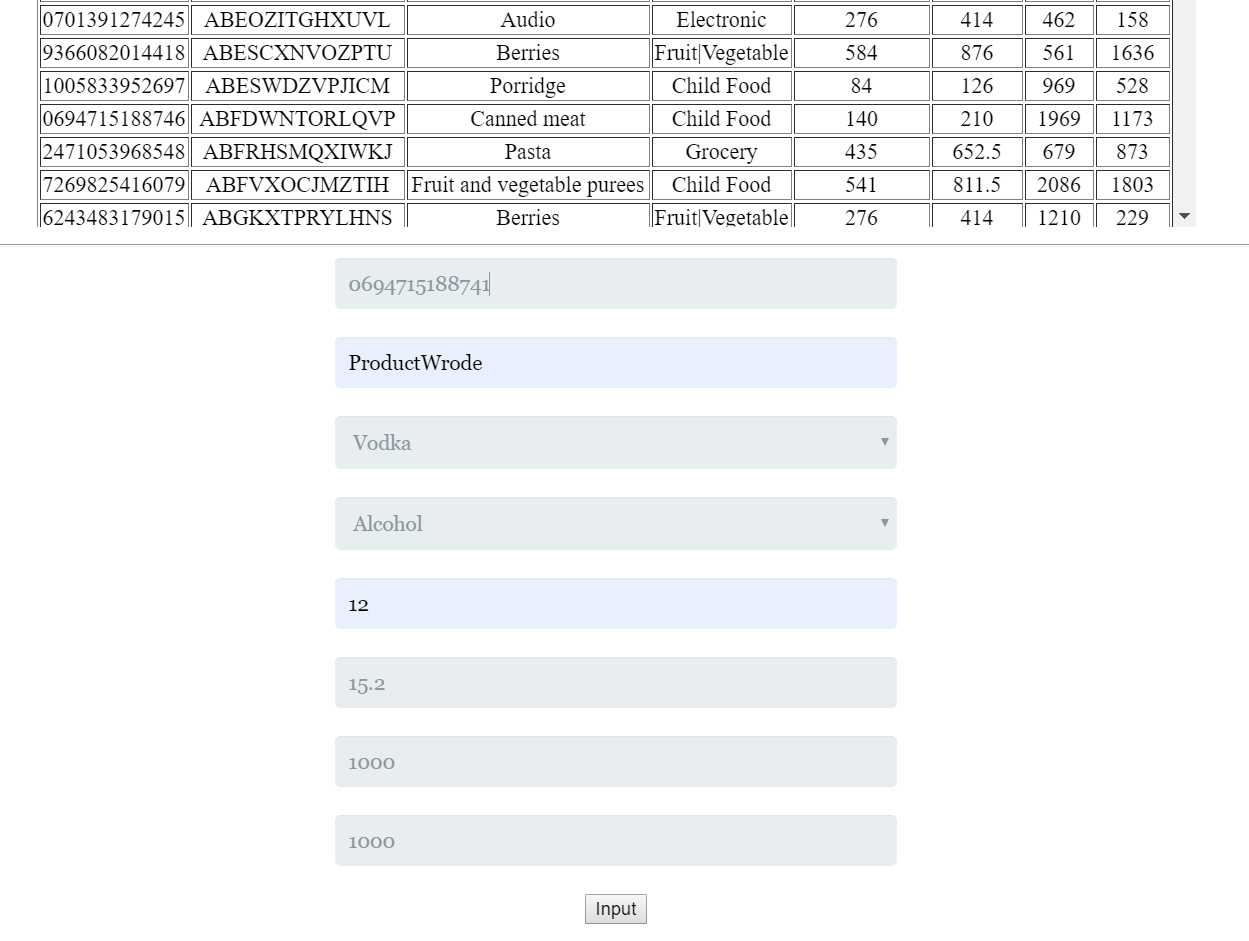


Рисунок 4.4 – Добавление товарной позиции

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.

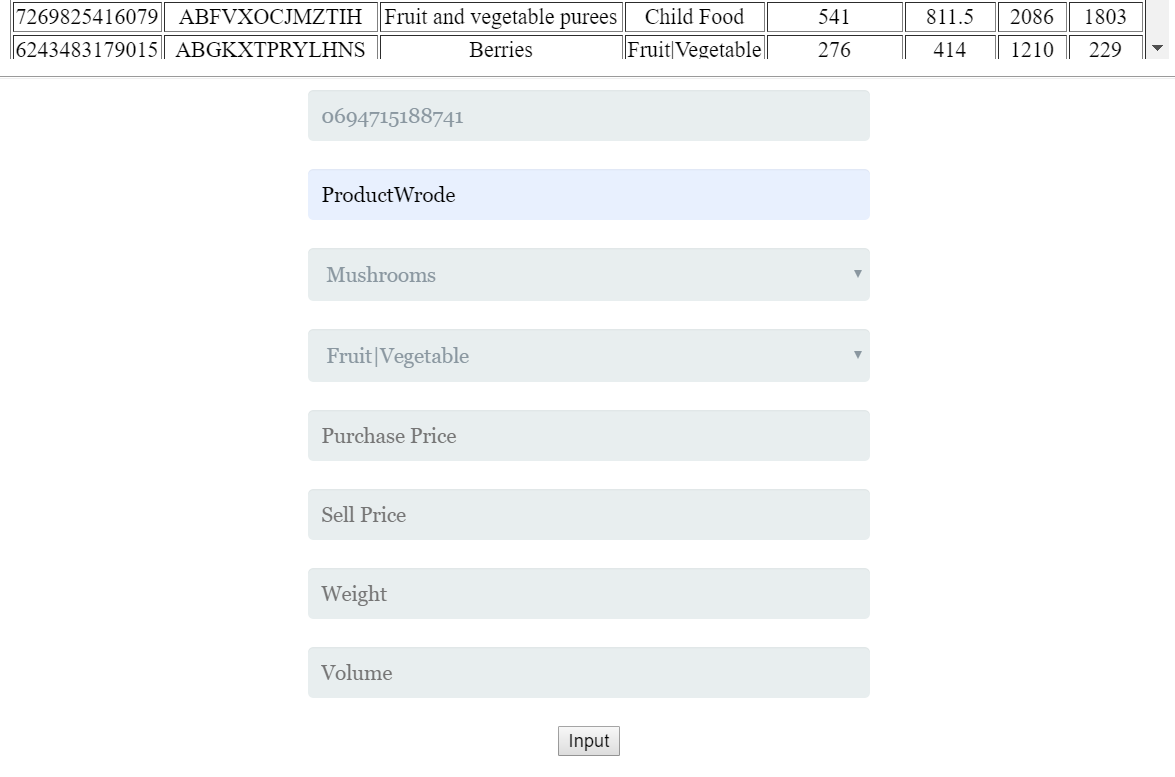


Рисунок 4.6 – Обновление товарной позиции

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

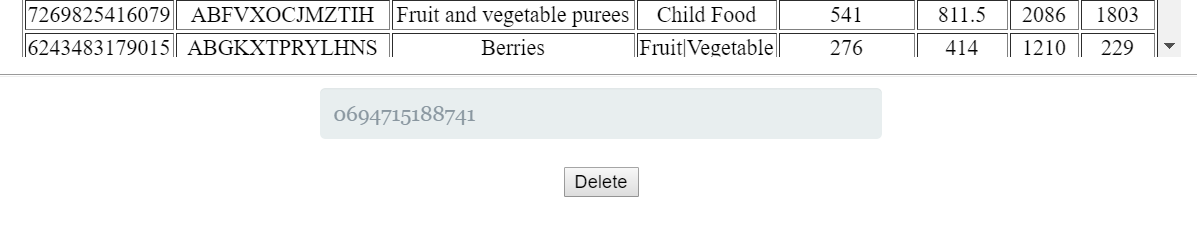


Рисунок 4.7 – Удаление товарной позиции

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.

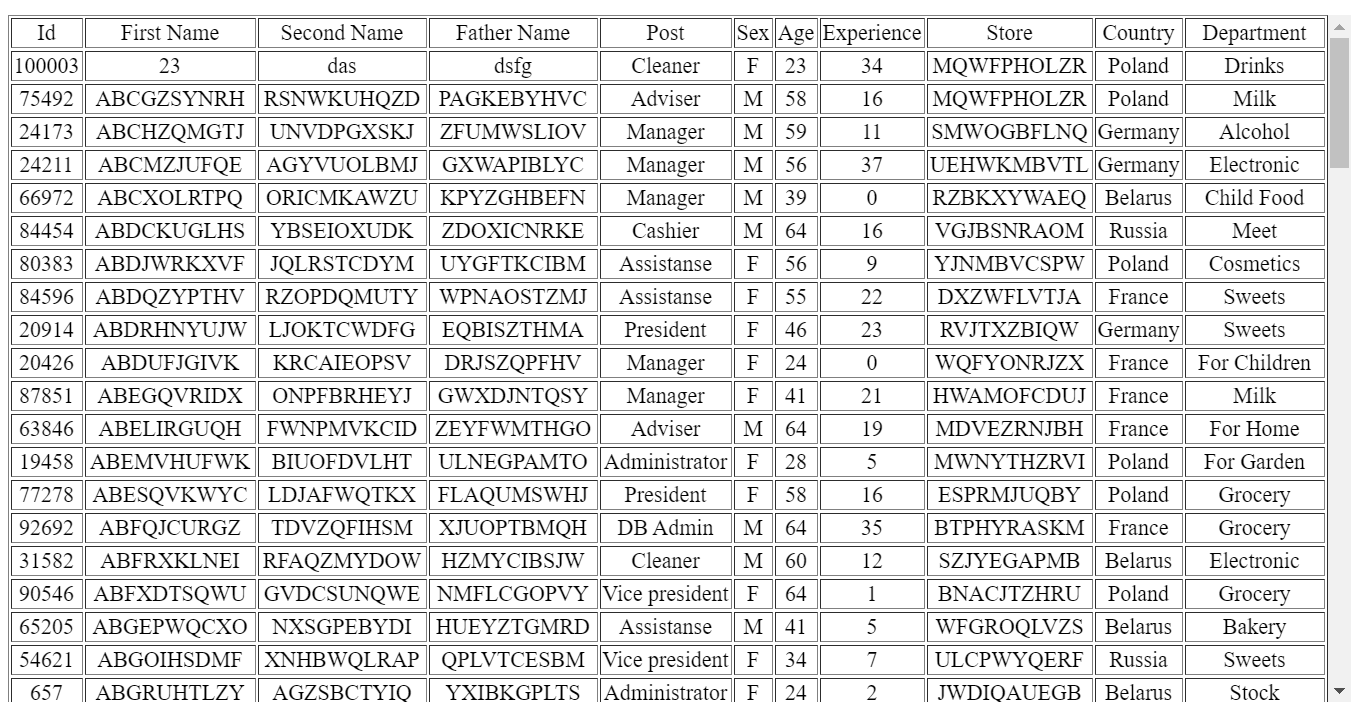


Рисунок 4.8 – Вывод списка работников

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

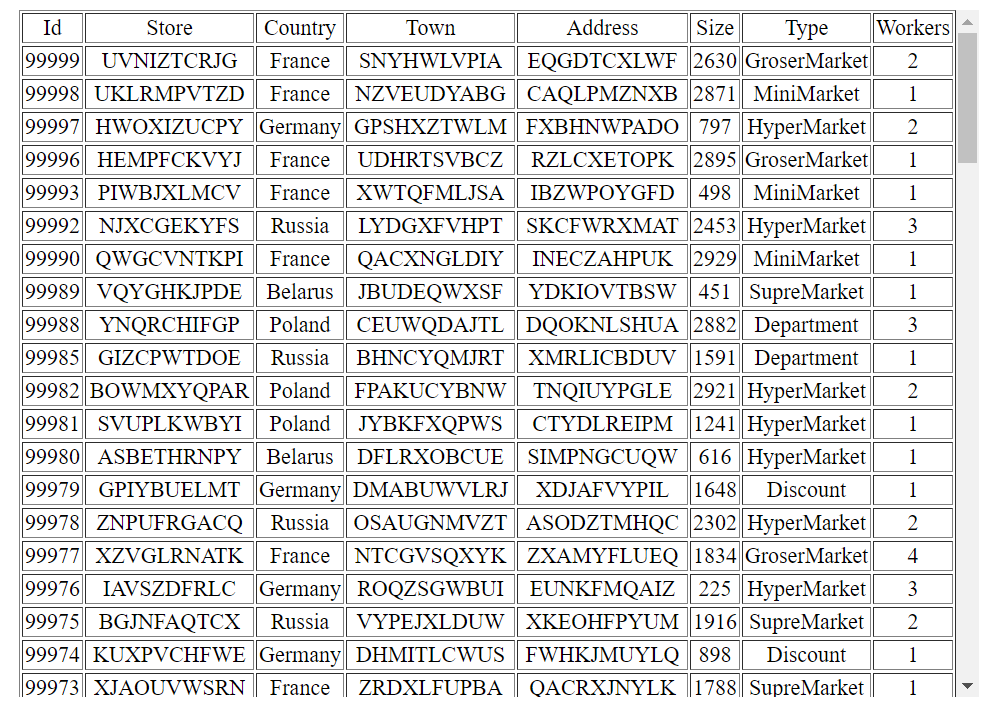


Рисунок 4.9 – Вывод списка магазинов

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

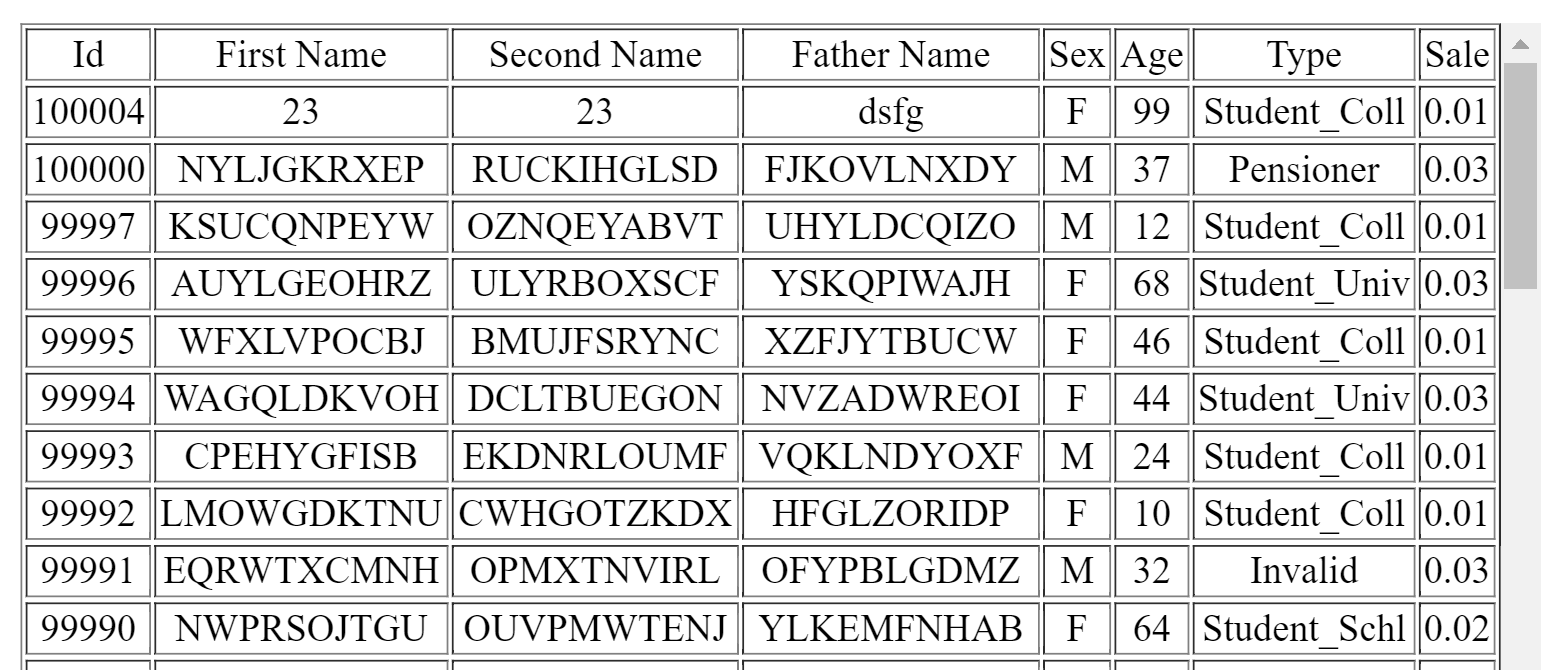


Рисунок 4.10 – Вывод списка покупателей

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

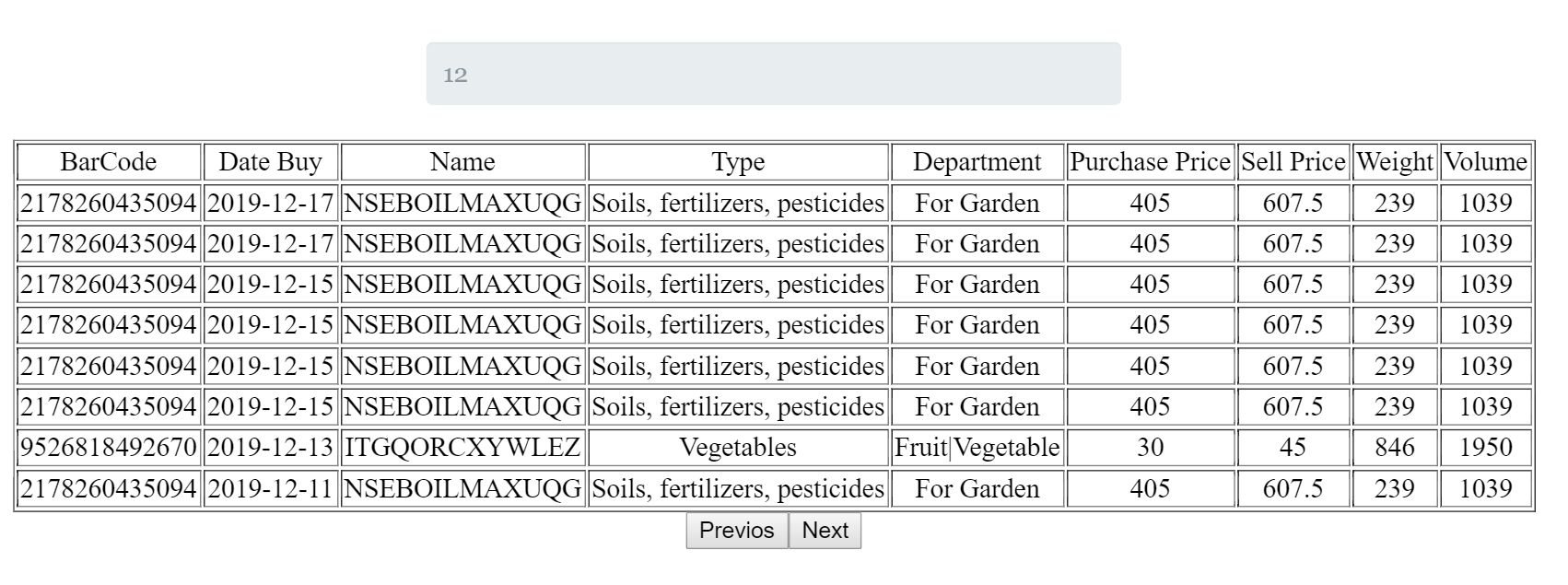


Рисунок 4.11 – Вывод списка покупок

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

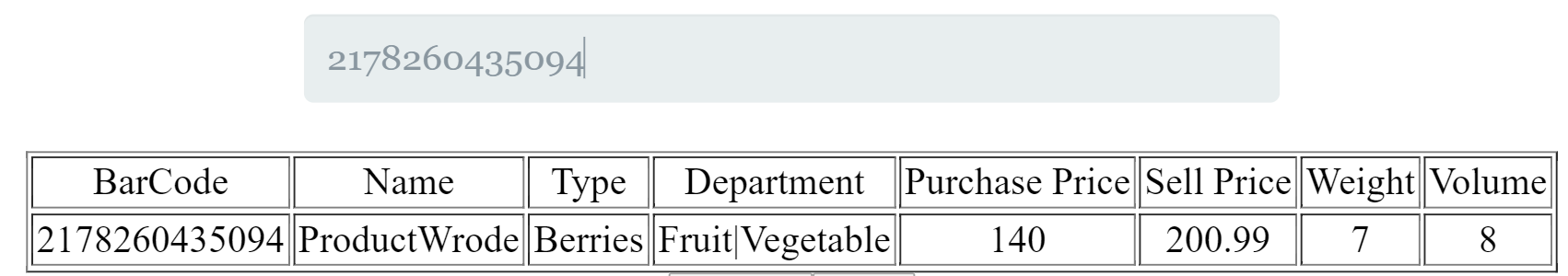


Рисунок 4.12 – Поиск товара по штрихкоду

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

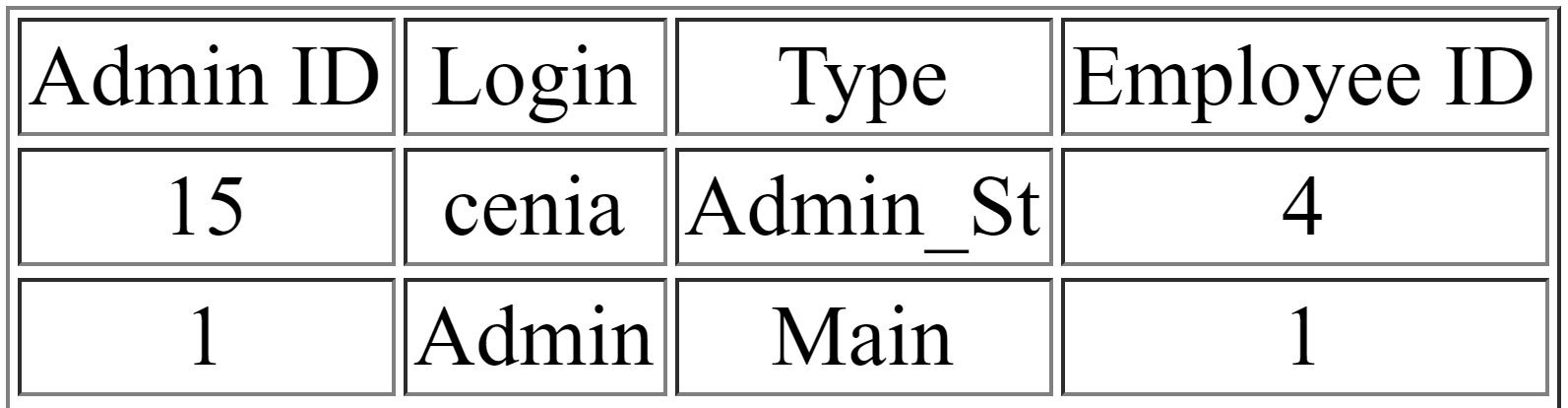


Рисунок 4.13 – Вывод списка администраторов

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

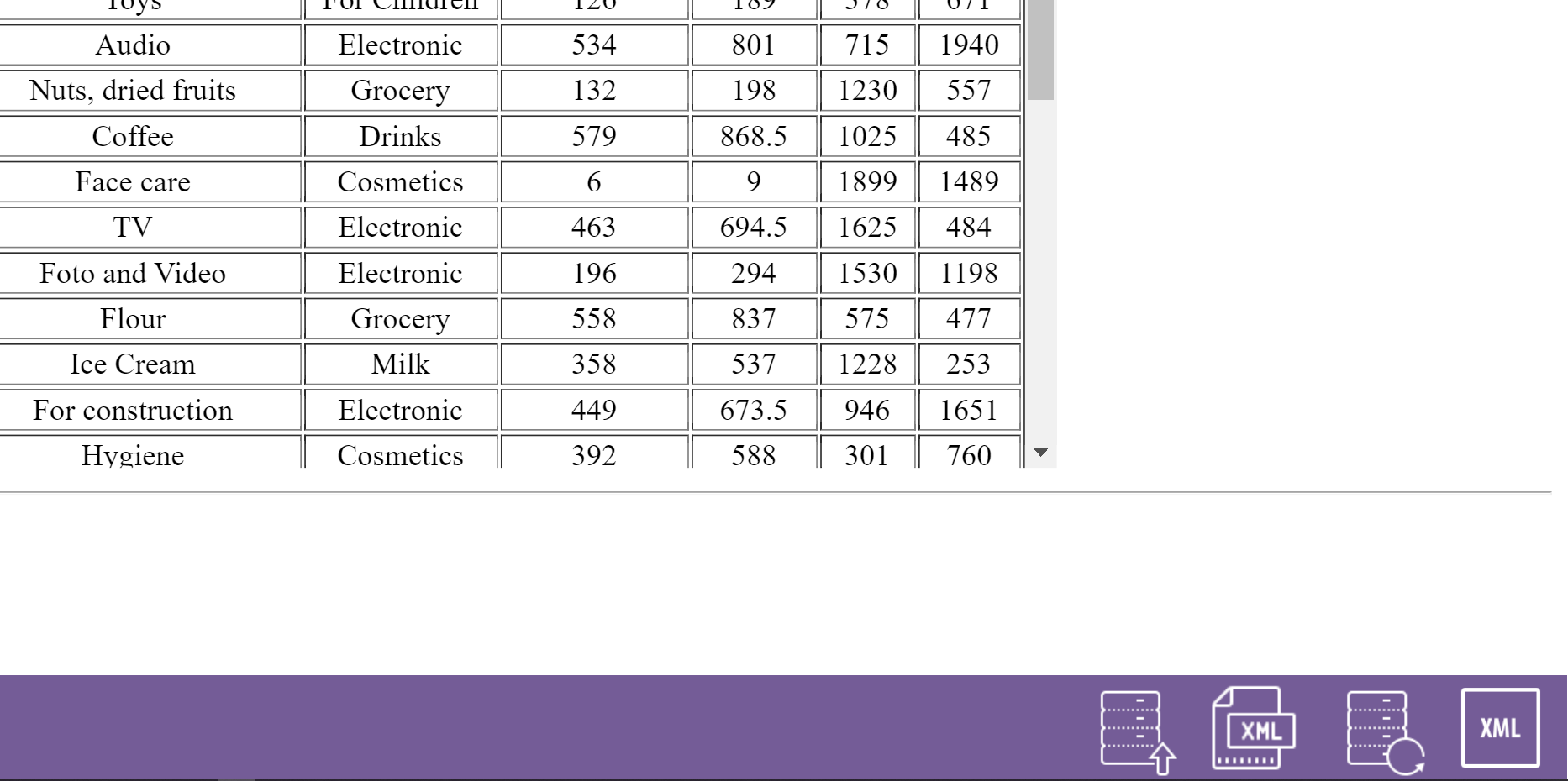


Рисунок 4.14 – Кнопки в нижнем углу

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

ФИО

Подпись

Дата

Лист

1

КП 00.00.ПЗ

Разраб.

Матюх А.А.

Провер.

Бурмакова А.В.

Н. контр.

Бурмакова А.В.

Утверд.

Бурмакова А.В.

Заключение

Лит.

Листов

1

БГТУ 71171075, 2019

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 взаимодействовие 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 объёме.

Список литературы

ФИО

Подпись

Дата

Лист

1

КП 00.00.ПЗ

Разраб.

Матюх А.А.

Провер.

Бурмакова А.В.

Н. контр.

Бурмакова А.В.

Утверд.

Бурмакова А.В.

Список литературы

Лит.

Листов

1

БГТУ 71171075, 2019

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.

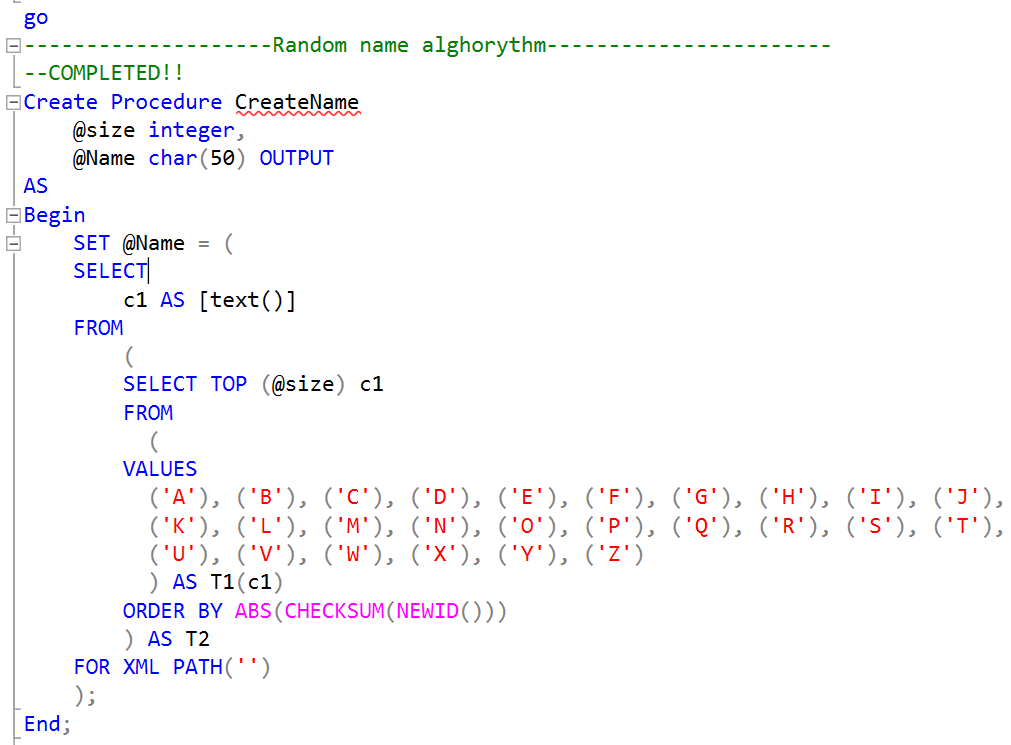
4. BACKUP (Transact-SQL) [Электронный ресурс] / Microsoft. – Режим доступа: https://docs.microsoft.com/en-us/sql/t-sql/statements/backup-transact-sql?view=sql-server-ver15. – Дата доступа: 04.11.2019.

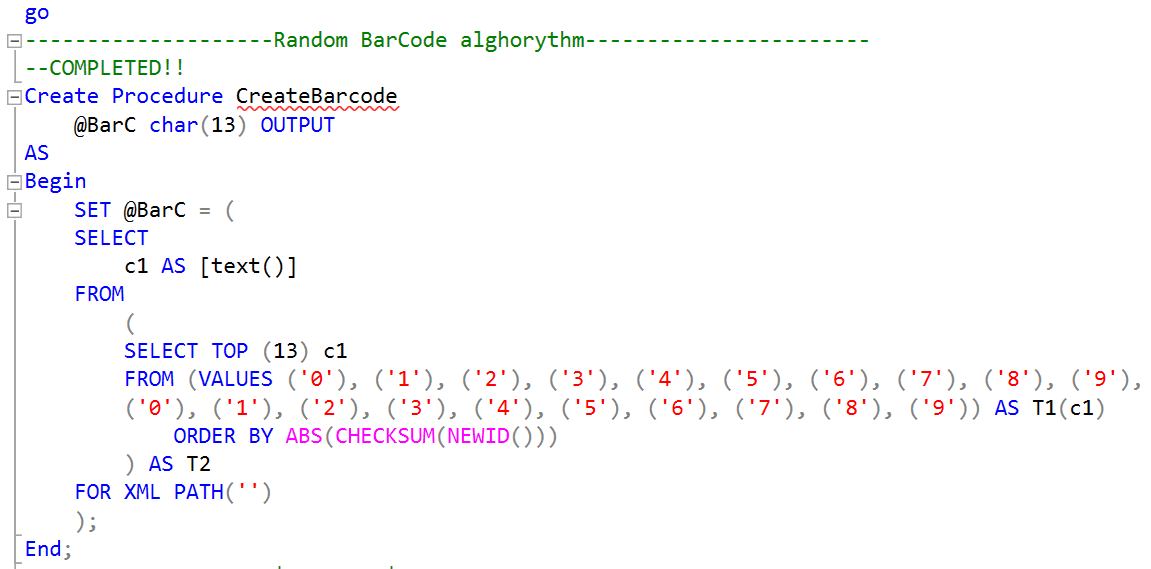
6. Restore a Database Backup Using SSMS [Электронный ресурс] / Microsoft. – Режим доступа: https://docs.microsoft.com/en-us/sql/relational-databases/backup-restore/restore-a-database-backup-using-ssms?view=sql-server-ver15. – Дата доступа: 15.11.2019.

Приложение А



Приложение Б





Приложение В

