Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Базы данных (БД)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

на тему

**Программное средство «Веб-приложение магазина музыкального оборудования»**

**БГУИР КР 1-40 01 01 624 ПЗ**

Студент: гр. 751006 Ситник А.А.

Руководитель:

асс. Фадеева Е.Е.

Минск 2020

Содержание

[**1 ВВЕДЕНИЕ** 3](#_Toc59055170)

[**2 Анализ прототипов, литературных источников и формирование требований к проектируемому программному средству** 4](#_Toc59055171)

[**3 АНАЛИЗ ТРЕБОВАНИЙ К ПС И РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ** 10](#_Toc59055172)

[**4 Инфологическая модель предметной области** 13](#_Toc59055173)

[**5 Подробное описание бизнес-логики** 14](#_Toc59055174)

[**5.1 Описание базы данных** 14](#_Toc59055175)

[**6 Тестирование базы данных и приложений системы** 15](#_Toc59055176)

[**ЗАКЛЮЧЕНИЕ** 16](#_Toc59055177)

[**СПИСОК ЛИТЕРАТУРЫ** 17](#_Toc59055178)

[**ПРИЛОЖЕНИЯ** 18](#_Toc59055179)

## **ВВЕДЕНИЕ**

Базы данных являются на сегодняшний день неотъемлемой частью любой сколь-нибудь крупной информационной системы любой направленности: начиная от широко распространённых веб-приложений (таких, как веб-приложения магазинов Интернет-торговли) и заканчивая широко распределёнными системами автоматического управления (например, применяемыми на железных дорогах).

Вне зависимости от предметной области, в рамках которой строится база данных, база данных должна иметь структуру, пригодную как для нормальной работы разработчиков, так и для удобного процесса её заполнения иными лицами.

Поэтому при разработке веб-приложения магазина музыкального оборудования

## **Анализ прототипов, литературных источников и формирование требований к проектируемому программному средству**

* 1. **Анализ прототипов**

Так как разрабатываемое программное средство будет представлять собой веб-приложение магазина музыкальных товаров, в качестве прототипов были выбраны сайты магазинов аналогичной тематики, такие как белорусский bysound.by, немецкий thomann.de, а также sweetwater.com из США.

2.1.1 Веб-приложение магазина bysound.by является классическим представителем данной области торговли: он разбит на разделы, соответствующие типам продукции (например, разделы «Гитары», «Клавишные», «Ударные»), которые выведены в специальное горизонтальное меню. На главной странице также представлены наиболее популярные по продажам продукты из всех разделов магазина, а также новые продукты и действующие в настоящее время акции.

К преимуществам данного приложения можно отнести следующее:

1. возможность добавлять страницы с заинтересовавшими потенциального покупателя товарами в раздел «Закладки» для последующего быстрого доступа к ним;
2. возможность провести сравнение двух товаров в том числе из разных категорий;

Недостатками являются:

1. скудное описание (вплоть до его отсутствия) и набор характеристик большинства товаров магазина. Так, к примеру, чехол для гитары может описываться лишь одной характеристикой – «Бренд». ­­­­­
2. неудачное графическое решение страницы отдельно взятого товара из-за смещения блока с изображениями товара влево относительно вертикальной линии симметрии;

2.1.2 Сайт магазина thomann.de являет собой пример лаконичного и вместе с тем функционального дизайна. К особенностям можно отнести присутствие на страницах практически всех разделов сайта (за исключением таких, как, например, «Оборудование для DJ») рекламы собственной продукции магазина под различными брендами.

Преимущества:

1. являясь международной сетью, предоставляет свои услуги на многих языках стран Европы, а также имеет возможность выбора валюты, использующейся в интерфейсе;
2. позволяет провести расчёт стоимости доставки товара без необходимости регистрации и аутентификации на сайте;
3. для большинства продуктов, представленных в магазине, имеется набор фотографий натурных образцов в количестве не менее 5 изображений с возможностью просмотра несжатого оригинала (с разрешением по большей стороне вплоть до 2000 пикселов.
4. наличие на страницах музыкальных инструментов так называемых «семплов» – примеров их звучания. Так для некоторых гитар количество таких записей может доходить до 8 единиц, что предоставляет потенциальному покупателю возможность более осознанно сделать выбор в пользу того или иного продукта;
5. возможность изучения мнения других покупателей на странице каждого товара с помощью блока, в котором указаны процентные соотношения пользователей сайта, выбравших данный продукт или продукцию конкурента;

К минусам можно отнести следующие пункты:

1. преобладание в рекомендациях товаров внутренних брендов компании, что может послужить отталкивающим факторов для покупателей, имеющих представление о данной области;
2. недостаточное количество фильтров продукции в режиме просмотра определённых категорий товаров. Например, при просмотре категории «Акустические гитары» есть опция фильтрации товаров по цене и бренду, однако нет возможности фильтровать товары по количеству струн (Примечание: своеобразным заменителем данной опции можно считать наличие отдельных разделов для инструментов с разным количеством струн, что, по мнению автора данной записки, является неудачным решением);
3. некорректная работа службы подбора связанных товаров: так, например, для звукового микшера с ценой N в разделе «Связанные товары» будут представлены продукты того же производителя, принадлежащие, однако, другим производственным сериям и отличающиеся по цене до 5N.

2.1.3 Веб-приложение магазина Sweetwater.com в целом повторяет удачные решения рассмотренного последним аналога, привнося некоторые отличительные черты.

К положительным особенностям относятся:

1. наличие для некоторых отдельных категорий и моделей товаров возможности просмотреть галереи с фотографиями всех образцов, имеющихся в наличии. К примеру, при наличии в магазине трёх экземпляров бас-гитары N в одной расцветке, на сайте будут представлены галереи изображений каждого из них с указанием в том числе и серийных номеров изделий;
2. доступ к достаточно подробному списку характеристик товаров (в среднем 20 пунктов для струнных музыкальных инструментов и 12 для звукозаписывающего периферийного оборудования).
3. для некоторых отдельных продуктов: возможность всеракурсного осмотра товара в окне браузера.

К негативным чертам стоит отнести следующие:

1. высокая нагрузка на графический ускоритель и центральный процессор устройства из-за большой плотности различных видов медиа (изображений, анимированных фонов, видеороликов) на страницах сайта;
2. дублирование категорий, выраженное в присутствии одной и той же подкатегории в разных более крупных.

* 1. **Анализ литературных источников**

В качестве литературных источников в процессе курсового проектирования использовались издания, перечисленные в разделе «Список литературы» данной записки. Краткий анализ их содержания приведён ниже в этом разделе.

1. *«Реляционные базы данных в примерах»* [1], Куликов С.С., 2019. Данное издание содержит основные понятия реляционных баз данных, систематизирует их в удобные категории и подкрепляет подаваемую информацию примерами на языке SQL разных диалектов (в их числе T-SQL, Oracle SQL, MySQL и другие).
2. «*Mastering SQL Server 2017: Build smart and efficient database applications for your organization with SQL Server 2017»* [2], Milos Radivojevic, 2019. В данной книге подробно изложены особенности как самой СУБД Microsoft SQL Server, так и принципы взаимодействия клиент-серверных приложений с базами данных, в частности, с использованием формата данных JSON. Большое внимание уделено использованию временных таблиц для оптимизации работы с базой данных.
3. «*SQL Server for Node.js*», Agus Kurniawan, 2013 [3]. Рассматриваемое издание ставит целью дать читателю методические рекомендации по использованию связки технологий «СУБД MS SQL – серверная часть приложения на фреймфорке Node.js».
4. Стандарт предприятия «Дипломные проекты (работы). Общие требования» [4], коллектив авторов, БГУИР, 2008. Данный стандарт содержит требования к оформлению работ студентов и преподавательского состава БГУИР.
5. *«Web Development with Node and Express: Leveraging the JavaScript Stack»*, Ethan Brown, 2014 [5]. Книга является практическим руководством по созданию веб-ориентированных приложений с использованием технологий Node.js в связке с фреймфорком Express, значительно упрощающим работу с Сетью (по аналогии с классами Java EE). Также рассмотрены принципы построения REST-архитектуры в веб-приложении;
6. *«SQL Cookbook: Query Solutions and Techniques for Database Developers»*, Anthony Molinaro, 2005. Представленное издание предназначено для разработчиков баз данных коммерческого уровня, содержит наиболее часто встречающиеся разработчикам проблемы и способы их решения;
7. *«SQL Server 2017 Developer's Guide: A professional guide to designing and developing enterprise database applications»*, Dejan Sarka, 2018. Автор рассматриваемой книги уделяет особое внимание отличительным особенностям диалекта языка SQL от Microsoft – Transact-SQL, предлагает методы оптимизации существующих баз данных или заново создающихся;
8. *«Learning Node.js»*, Marc Wandschneider, 2016. Книга посвящена углубленному изучению платформы Node.js как средства разработки веб-приложений, а также последующему развёртыванию разработанного программного средства. Этап разработки описан в ней вплоть до конкретных алгоритмов взаимодействия с источниками данных (сюда относятся и базы данных ;
9. *«SQL Antipatterns: Avoiding the Pitfalls of Database Programming»*, Bill Karwin, 2017. Издание целиком состоит из разбора так называемых «антипаттернов», то есть типичных ошибок начинающих и профессиональных программистов баз данных, которые оказывают значительное влияние на производительность итогового продукта и усложняют процесс менеджмента данных;
10. *«Database Design for Mere Mortals: A Hands-On Guide to Relational Database Design»*, Michael Hernandez, 2018. Книга представляет собой справочник по реляционным базам данных (в том числе MS SQL Server), предлагает различные примеры реализации базовых функций и процедур, а также содержит перечень наиболее часто встречающихся ошибок при работе с базами данных, причины их появления и способы устранения.
    1. **Формирование требований к проектируемому программному средству**

2.3.1 Назначение разработки и состав выполняемых функций (требования)

Проектируемое программное средство должно обеспечивать нормальное функционирование магазина, специализирующегося на продаже музыкального оборудования, предоставляя его сотрудникам средства для работы с базой продаваемых товаров (базой данных системы), клиентской базой (управление заказами в ручном и автоматическом режимах), сектором материального обеспечения (контроль за наполненностью складских помещений), а клиентам – возможность ознакомления с ассортиментом товаров магазина и последующего совершения покупки с оплатой через интегрированную в приложение систему оплаты.

Ниже представлено более развёрнутое описание требований к проектируемому программному средству.

Данное приложение должно предоставлять зарегистрированным пользователям (потенциальным покупателям) следующие возможности:

* осуществлять просмотр всех доступных товаров в магазине.
* сравнивать товары из одной категории.
* проверить, есть ли конкретный товар в наличии.
* осуществить заказ товаров через раздел «Корзина».

Незарегистрированным посетителям сайта должны быть предоставлены следующие функции:

* просмотр ассортимента магазина с возможностью его сортировки и фильтрации по параметрам, характерным для категории конкретного товара;
* сохранение понравившегося товара в разделе «Понравившееся» с возможностью последующего сохранения после аутентификации;

Для группы пользователей, представляющих сотрудников магазина, после успешной аутентификации должны быть реализованы следующие функции:

* добавление/удаление/редактирование информации о товарах.
* просмотр статистики продаж за фиксированный срок (день, неделя, месяц, год).
* поддержание в актуальном состоянии информации о наличии товаров на складе (-ах).
* просмотр информации о заказах.
* подтверждение или отклонение заказов в ручном режиме.

2.3.2 Требования к программной совместимости

В соответствии с тем, что законченный проект будет представлять собой веб-приложения с клиентской частью, предъявляются определённые требования к поддерживаемым веб-браузерам для просмотра страниц.

Клиентская часть веб-приложения должна поддерживать следующие браузеры:

* Mozilla Firefox не старше версии 72;
* Google Chrome от версии 58 и выше;

Для нормального функционирования серверной части приложения необходима серверная операционная система Microsoft Windows Server версии не старше 2010 с интегрированным в неё фреймворком Node.js.

2.3.3 Постановка задачи

В результате выполнения курсового проектирования должно быть разработано программное средство – веб-приложение, связанное с базой данных магазина музыкального оборудования, – которое будет доступно пользователям сети Интернет через браузер. Программное средство должно быть защищено с помощью процедур аутентификации пользователей, а также сетевого протокола HTTPS.

## **АНАЛИЗ ТРЕБОВАНИЙ К ПС И РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ**

* 1. **Обоснование выбора СУБД и иных средств разработки**

Разработка проекта будет вестись на основе следующих технологий: серверная часть веб-приложения – на платформе Node.js с фреймворком Express.js; система управления базой данных – Microsoft SQL Server (диалект T-SQL; в дальнейшем – MS SQL); клиентская часть должна иметь в основе разметку на языке HTML/CSS/Javascript. Ниже по пунктам изложены преимущества и недостатки выбранных инструментов разработки:

1. Платформа Node.js.

Преимущества:

1. упрощение процесса разработки клиент-серверного приложения за счёт унификации используемых языков программирования (Javascript);
2. наличие собственного менеджера пакетов – npm (от «Node Package Manager»), который используется для установки расширений платформ (пакетов);
3. доступность фреймворка Express.js – де-факто стандарта при разработке веб-приложений, который предоставляет разработчику более удобную работу с маршрутизацией запросов через использование так называемых «рутеров», то есть сущностей, осуществляющих контроль за запросами;
4. возможность использовать различные реализации технологии веб-сокетов (в данном проектировании – socket.io), которые позволяют изменить сам подход к разработке клиент-серверных приложений, когда обе части становятся равноправными участниками процесса обмена информацией. То есть при использовании данного пакета отправлять запросы и получать ответы может не только клиентская часть, но и серверная в соответствии со своими потребностями;
5. поддержка так называемых «движков представлений» – аналогов JSP-страниц в Java EE – которые позволяют встраивать Javascript-код напрямую в код HTML-страниц или наоборот, что даёт возможность упростить структуру приложения за счёт избавления от отдельных скриптовых файлов, которые отправлялись бы на клиент;
6. возможность подключения к базам данных (в том числе Microsoft SQL Server, о котором будет написано ниже) через специальные объекты-обёртки, которые подключаются к проекту как зависимости (например, пакет «mssql»).

Недостатки:

1. те же пакеты npm могут иметь негативный фактор воздействия, так как они, будучи включёнными в проект, увеличивают размер проекта, а также могут иметь уязвимости;
2. отсутствие нативной поддержки многопоточности, её использование возможно только за счёт установки npm-пакетов, однако и они лишь эмулируют многопоточность. Причиной этого недостатка является одна из ключевых особенностей самого языка javascript, а именно однопоточность, выгодная для браузеров, но не для серверов;
3. зависимость используемых в приложении библиотек от сетей их распространения (CDN – Content Delivery Network). Так, удаление разработчиком пакета последнего из библиотеки пакетов теоретически может привести к потере работоспособности приложениями, его использующими.
4. СУБД Microsoft SQL Server (включая среду разработки Microsoft SQL Management Studio):

Преимущества:

1. наличие хорошо задокументированных особенностей диалекта языка запросов Transact-SQL (далее T-SQL), в том числе в онлайн-документации [11];

В качестве основной среды разработки была выбрана интегрированная среда разработки JetBrains WebStorm 2019 Enterprise Edition. К её преимуществам можно отнести следующие пункты:

**3.2 Формирование функциональных требований к ПС**

3.2.1 Пользователи системы и их роли

В проектируемой системе можно выделить три роли для пользователей:

1. незарегистрированный пользователь;
2. зарегистрированный пользователь;
3. сотрудник магазина.

Роль «сотрудник магазина» в свою очередь может быть подразделена для более точного описания должностных обязанностей, однако в масштабах рассматриваемой системы выделение отдельных ролей для, например, менеджеров по продажам или менеджеров по связям с клиентами.

3.2.2 Функциональные требования к программному средству

На основании требований, выделенных в предыдущем разделе, сформируем функциональные требования к разрабатываемому программному средству, разбив их по ролям пользователей будущего продукта.

1. Доступные для всех пользователей системы (в том числе незарегистрированных):
   1. при переходе на главную страницу на последней отображаются наиболее продаваемые за последнюю неделю продукты из всех категорий без какой-либо дополнительной сортировки;
   2. на каждой странице сайта будет доступен выбор валюты, в которой будут отображаться цены (справочно);
   3. в интерфейсе страницы присутствует кнопка «Понравившееся», по нажатию на которую пользователь сможет перейти в соответствующий раздел, где будут сохранены товары;
   4. с помощью графических элементов типа «checkbox» доступно добавление товаров в список для сравнения с возможностью последующего просмотра;
   5. возможен просмотр отдельной категории товаров: например, только акустических гитар (все бренды), только синтезаторы (все бренды) с возможностью фильтрации и сортировки;
2. Функции приложения, доступные исключительно сотрудникам магазина:
   1. просмотр статистики по продажам за выбранный срок (текущий день, прошедшая неделя, месяц или год);
   2. просмотр статистики по количеству новых пользователей;
   3. управление состоянием таблиц баз данных, касающихся информации о наличии товара на определённом складе (складской учёт);
   4. обработка заказов пользователей в ручном режиме, подтверждение или отклонение их, при необходимости редактирование;
   5. возможность создания сущности заказа в БД вручную (например, для ситуации, когда потенциальный покупатель оставляет заказ не через сайт, а с помощью звонка по телефону);
   6. актуализация ассортимента продукции магазина с помощью выставления статуса «Больше не выпускается» / «Retired» для устаревшего товара (соответствующие изменения вносятся в базу данных);
3. Функции программного средства, которые доступны зарегистрированным пользователям:
   1. просмотр содержимого вкладки «Корзина», его сохранение даже после завершения текущей сессии;
   2. оформление заказа на товары, находящиеся в разделе «Корзина»;
   3. управление совершёнными (в том числе неподтверждёнными магазином) заказами: отмена, редактирование его подробностей.
   4. **Особенности взаимодействия частей клиент-серверного приложения**

Для взаимодействия клиентской части веб-приложения и серверной была выбрана технология веб-сокетов (конкретная реализация – библиотека *socket.io* для языка Javascript).

Главной особенностью данной технологии является то, что она преобразует ранее синхронный процесс обмена запросами и ответами, типичный для стандартного сетевого протокола HTTP/HTTPS, на асинхронный. Это означает, что при использовании веб-сокетов не только клиент может отправлять запросы на сервер и получать ответ от последнего, но и сервер может по собственной необходимости посылать запрос (возможно, без ожидаемой возвращаемой информации) на клиент. Таким образом, сервер и клиент становятся равноправными участниками процесса обмена информацией.

Так, например, при переходе на главную страницу запрос к базе данных (что очень важно, также *асинхронный* в языке Javascript) выполняется сервером после получения соответствующего запроса со стороны клиента.

Процесс обновления базы данных сотрудником завершается тем, что серверное приложение при помощи сокетов отправляет всем своим подключениям последние изменения, произведённые на сервере базы данных. Тем самым решается проблема рассинхронизации состояния веб-приложения у разных клиентов, которая при применении «чистого» протокола HTTP могла быть исправлена лишь через обновление страницы и, соответственно, отправку повторного запроса на сервер.

## **Инфологическая модель предметной области**

В данном разделе будут подробно рассмотрены сущности, используемые в проектируемой базе данных, а также связи между ними.

* 1. **Сущности базы данных**

4.1.1 «Ядром» всей базы данных можно назвать сущности, соответствующие категориям реальных товаров: «Акустические гитары», «Фортепиано», «Микрофоны» и другие. Ниже в этом подразделе приведены словесные описания лишь некоторых из этих сущностей. Логика выбора сущностей для описания заключается в том, что для нескольких сущностей, представляющих разные, но «родственные» объекты реального мира, описывается та, большая часть атрибутов которой является общей для его группы товаров.

Каждый тип товара в качестве первичного ключа содержит уникальный идентификатор в виде Unicode-строки размеров 7 символов. Формат строки следующий: первые три символа являются зашифрованным обозначением типа товара/категории (например, электрогитары шифруются сочетанием «ELG», барабаны – «DRM»), остальные четыре символа – порядковый номер записи по очерёдности её добавления в базу данных.

4.1.2 Сущность *«Акустическая гитара»* соответствует реальному объекту – акустической гитаре. Набор атрибутов для данной сущности:

* уникальный идентификатор (строка);
* наименование (строка);
* количество струн (натуральное число);
* калибр (толщина) верхней струны в дюймах;
* калибр нижней струны в дюймах;
* мензура инструмента (длина звучащей части струны, традиционно измеряется в миллиметрах);
* материал грифа инструмента;
* материал деки (корпуса);
* материал накладки на гриф;
* материал накладки деки (может отсутствовать).

**4.2 Диаграммы сущностей и их связей**

## **Подробное описание бизнес-логики**

## **Описание базы данных**

* + 1. Представления

Так как в спроектированной базе данных каждый тип товаров представлен отдельными сущностями (таблицами), для удобства работы персонала, занимающегося обслуживанием базы данных, было создано представление dbo.vProducts, объединяющее в себе все актуальные (то есть не находящиеся в таблице dbo.RetiredProducts) товары. Общими для товаров разных категорий будут являться столбцы «ProductID» с уникальным идентификатором товара, «Name» с названием товара, а также столбец «Description» с описанием позиции.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя столбца | ProductID | Name | Description | Price |
| Тип данных | int | nvarchar(50) | nvarchar(250) | money |

Структура представления dbo.vProducts

* + 1. Триггеры

## **Тестирование базы данных и приложений системы**

## **ЗАКЛЮЧЕНИЕ**

В результате выполнения курсового проектирования были изучены и применены на практике принципы построения базы данных для определённой предметной области («Магазин музыкального оборудования»), разработано веб-приложение, которое использует построенную базу данных, предоставляя различные методы для взаимодействия с последней (например, добавление и удаление информации, формирование отчётов и прочее). авы

## **СПИСОК ЛИТЕРАТУРЫ**

[1] Куликов С.С. Реляционные базы данных в примерах / С. Куликов – Мн.: «ЭПАМ Системз», 2019 – 422 с.

[2] Mastering SQL Server 2017: Build smart and efficient database applications for your organization with SQL Server 2017 / Milos Radivojevic, Packt Publishing. – Birmingham, UK, 2019.

[3] SQL Server for Node.js / Agus Kurniawan, PE Press. – Inland Empire, Ca, 2013.

[4] Стандарт предприятия. Дипломные проекты (работы). Общие требования / коллектив авторов. – Мн.: БГУИР, 2008 – 171 с.

[5] Web Development with Node and Express: Leveraging the JavaScript Stack / Ethan Brown, O’Reilly. – Sebastopol, Ca, 2014.

[6] SQL Cookbook: Query Solutions and Techniques for Database Developers / Anthony Molinaro, O’Reilly. – Sebastopol, Ca, 2005.

[7] SQL Server 2017 Developer's Guide: A professional guide to designing and developing enterprise database applications / William Durkin, Packt Publishing. – Birmingham, UK, 2018.

[8] Learning Node.js / Marc Wandschneider, Addison-Wesley Professional. – Boston, MA, 2016.

[9] SQL Antipatterns: Avoiding the Pitfalls of Database Programming» / Bill Karwin, Pragmatic Bookshelf. – Boston, MA, 2010.

[10] Database Design for Mere Mortals: A Hands-On Guide to Relational Database Design / 3rd Edition / Michael Hernandez et al., Addison-Wesley Professional. – Boston, MA, 2013.

[11]

## **ПРИЛОЖЕНИЯ**

* + 1. Дамп базы данных