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 курсовому проекту

на тему

База данных для книжного интернет-магазина

БГУИР КП I–40 01 01 05 ПЗ

Студент: гр. 051003 Гаркушенко С.С.

Руководитель: асс. Колотыгин К. Е.

Минск 2023

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

УТВЕРЖДАЮ

Заведующий кафедрой ПОИТ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Лапицкая Н.В. 2023г.

ЗАДАНИЕ

по курсовому проектированию

Студенту Гаркушенко Станиславу Сергеевичу

1. Тема работы База данных книжного интернет-магазина

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Срок сдачи законченной работы 05.12.2023г.

3. Исходные данные к работе СУБД Oracle. Должен быть предусмотрен язык, жанр, автор для книг. Пользователь может добавлять книги в избранное и совершать заказ. Для заказа необходимы адресные данные. Просматривать историю заказов.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Содержание расчетно-пояснительной записки (перечень вопросов, которые подлежат разработке)

Введение

1. Анализ прототипов и формирование требований к программному средству

2. Анализ и разработка функциональных требований

3. Инфологическая модель предметной области

4. Схема базы данных

5. Подробное описание бизнес-логики

6. Тестирование и проверка работоспособности программного средства

Список использованной литературы

5. Перечень графического материала (с точным обозначением обязательных чертежей и графиков)

Схема данных приложения для интернет-магазина книг, А1, чертеж

6. Консультант по курсовой работе Колотыгин К. Е.

7. Дата выдачи задания 03.09.2023г.

8. Календарный график работы над проектом на весь период проектирования (с обозначением сроков выполнения и процентом от общего объема работы):

Раздел 1. Введение к 15.09.2023 – 15 % готовности работы;

Разделы 2, 3 к 15.10.2023 – 30 % готовности работы;

Разделы 4, 5 к 15.11.2023 – 60 % готовности работы;

Раздел 6. Заключение к 03.12.2023 – 90 % готовности работы;

Оформление пояснительной записки и графического материала к 05.12.2023 – 100 % готовности работы.

Защита курсового проекта с 05.12.2023г. по 18.12.2023г.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

РУКОВОДИТЕЛЬ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_К. Е. Колотыгин

(подпись)

Задание принял к исполнению С.С. Гаркушенко 03.09.2023г.

(дата и подпись студента)

Содержание

[Введение 5](#_Toc153272554)

[1 Анализ прототипов и формирование требований к программному средству 6](#_Toc153272555)

[1.1 Анализ существующих прототипов 6](#_Toc153272556)

[1.2 Формирование требований к программному средству 9](#_Toc153272557)

[2 Анализ и разработка функциональных требований 10](#_Toc153272558)

[2.1 Используемые технологии при разработке 10](#_Toc153272559)

[2.2 Функциональные требование к программному средству 11](#_Toc153272560)

[3 Инфологическая модель предметной области 12](#_Toc153272561)

[3.1 Описание предметной области 12](#_Toc153272562)

[3.2 Модель данных 12](#_Toc153272563)

[4 Схема базы данных 16](#_Toc153272564)

[5 Подробное описание бизнес-логики 22](#_Toc153272565)

[5.1 Триггеры 22](#_Toc153272566)

[5.2 Представления 22](#_Toc153272567)

[6 Тестирование и проверка работоспособности программного средства 23](#_Toc153272568)

[6.1 Триггеры 23](#_Toc153272569)

[6.2 Представления 26](#_Toc153272570)

[Заключение 28](#_Toc153272571)

[Список использованной литературы 29](#_Toc153272572)

Введение

Одним из направлений, стремительно развивающихся в современной экономике, в том числе и в Республике Беларусь, является электронная коммерция. С точки зрения технического потенциала белорусский интернет готов к развитию электронной коммерции. Количество пользователей и абонентов сети Интернет в нашей стране на 2017 г. составляло 11,083 млн. В 2017 г. на 100 человек населения количество абонентов сети составило 1170 человек, а по сравнению с 2012 г. (880 человек) этот показатель вырос на 33 % [1], что составляет действительно значительное приращение.

Купля-продажа товаров в секторе розничной онлайн-торговли занимает достаточно заметное место среди различных видов экономической деятельности в сети Интернет. По состоянию на 1 июля 2017 г., в Торговом реестре Республики Беларусь зарегистрировано 13811 интернет-магазинов. Прирост количества зарегистрированных интернет-магазинов в 2016 г. составил 22,8 %, или 2565 интернет-магазинов [1].

Интернет-магазин — это специализированный сайт, с помощью которого можно продавать и покупать товары и услуги в интерактивном режиме, предварительно ознакомившись с информацией о них. Само понятие «Интернет-магазин» возникло в 1994 г. Американец Джефф Безос был первым, кто предложил делать покупки в Интернете [2].

К преимуществам функционирования Интернет-магазинов можно выделить следующие: легкость и удобство в поиске товара, широкий ассортимент товаров, круглосуточный режим работы магазина, отсутствие очередей, доставка товара в любое место и время. К недостаткам Интернет-магазинов относят невозможность подержать товар в руках, необходимость оставлять свои личные данные при заказе товара, проблемы качества доставленного товара и безопасности платежей.

Вышеупомянутые тенденции в экономической и предпринимательской деятельности современного белорусского общества не могли не оказать влияние и на книготорговлю, букинистику. Общее снижение книготорговли мотивирует книжных издателей искать инновационные средства и методы для привлечения новой и большей читательской аудитории. Существует значительное количество примеров успешного использования современных технологий в книготорговле, это книжные интернет-магазины с автоматизированной системой платежей и доставки (каталог книг на ozon.ru, oz.by и amazon.com, ibook.com.ua, mybooks.by, bookdepository.com, chitatel.by, biblio.by, bookland.by и др.) [3].

Цель курсового проектирования – создание базы данных интернет-магазина книг с использованием современных принципов системного проектирования, глубинного анализа заданной предметной области с целью достижения и обеспечения максимально положительного опыта пользователя.

# Анализ прототипов и формирование требований к программному средству

## Анализ существующих прототипов

Был проведен сравнительный анализ следующих интернет-магазинов: oz.by, ozon.ru и mybooks.by.

При детальном изучении было выявлено, что книги чаще всего делят по категориям, издателям, жанрам. Помимо этого, существует деление на электронные книги, книги с твердой и мягкой обложкой, аудиокниги и проч.

### Веб-приложение oz.by

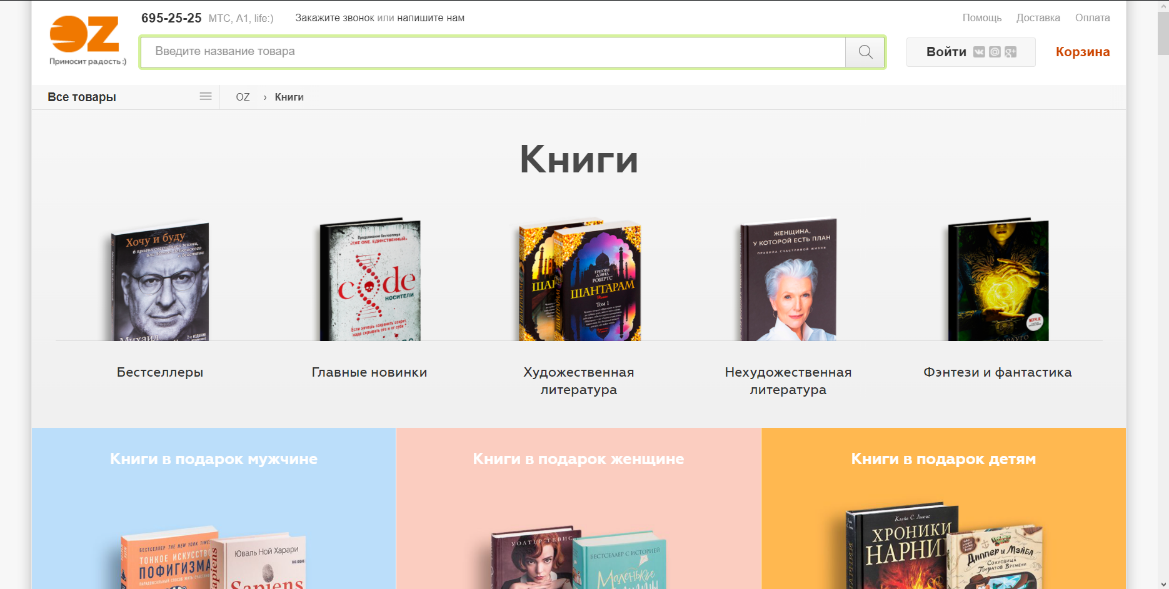


Рисунок 1.1 – Интерфейс веб-приложения oz.by

Рисунок 1.1 иллюстрирует интерфейс веб-приложения oz.by – это один из наиболее популярных сервисов для покупки книг в Республике Беларусь. Из функций можно выделить: гибкий поиск книг, распределение книг по жанрам и авторам.

Плюсы:

* поиск книг по жанрам;
* гибкий поиск книг;
* дисконтная программа;

Минусы:

* устаревший интерфейс;

### Веб-приложение ozon.ru

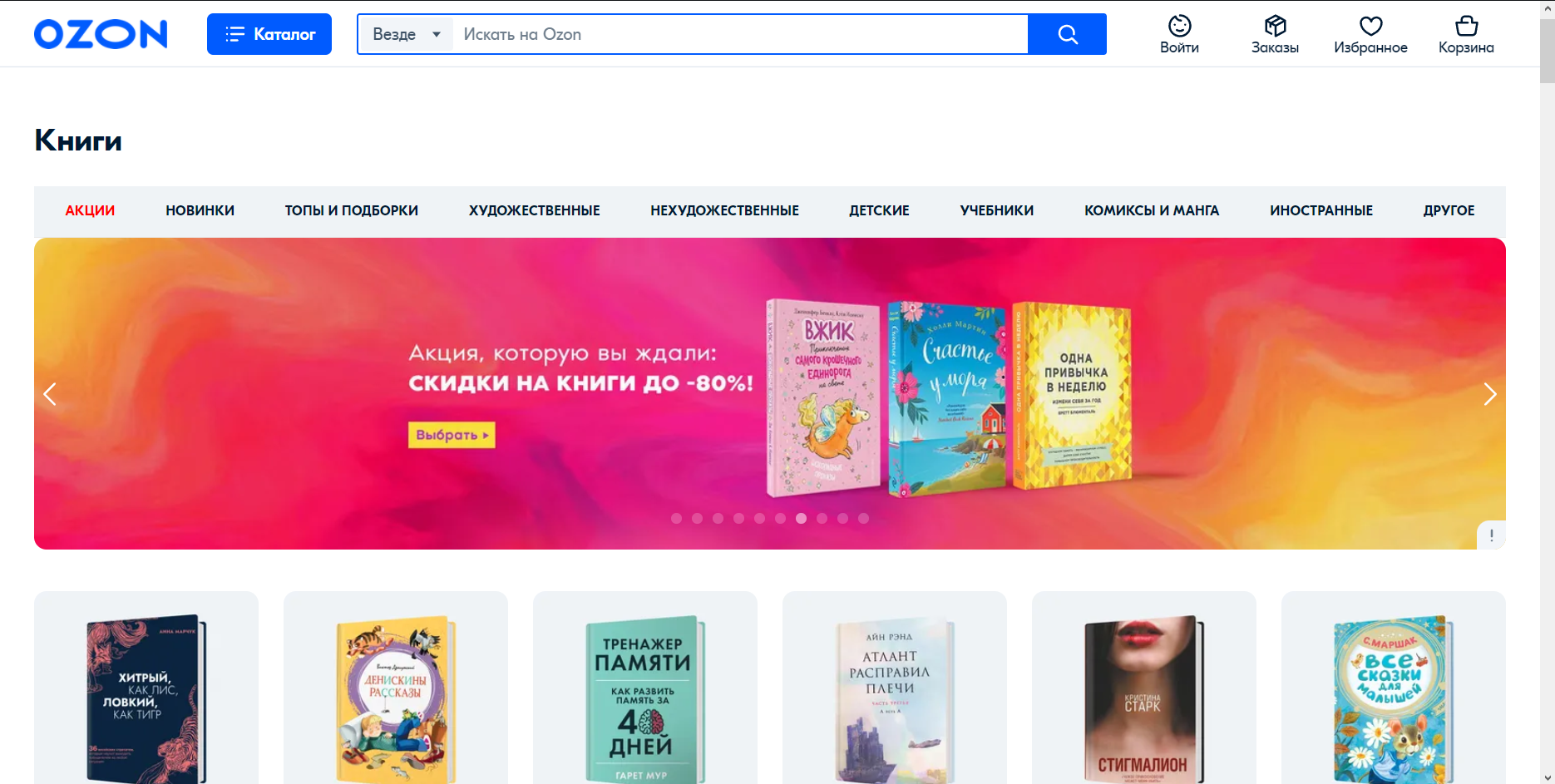


Рисунок 1.2 – Интерфейс веб-приложения ozon.ru

На рисунке 1.2 представлен интерфейс веб-приложения ozon.ru. Из функций можно выделить: гибкий поиск книг, наличие большого количества товаров кроме книг, начисление компенсаций и бонусов.

Плюсы:

* гибкий поиск книг;
* наличие большого количества товаров кроме книг;
* начисление компенсаций и бонусов;

Минусы:

* отсутствие возможности отправки документов, чеков покупателям.

### Веб-приложение mybooks.by

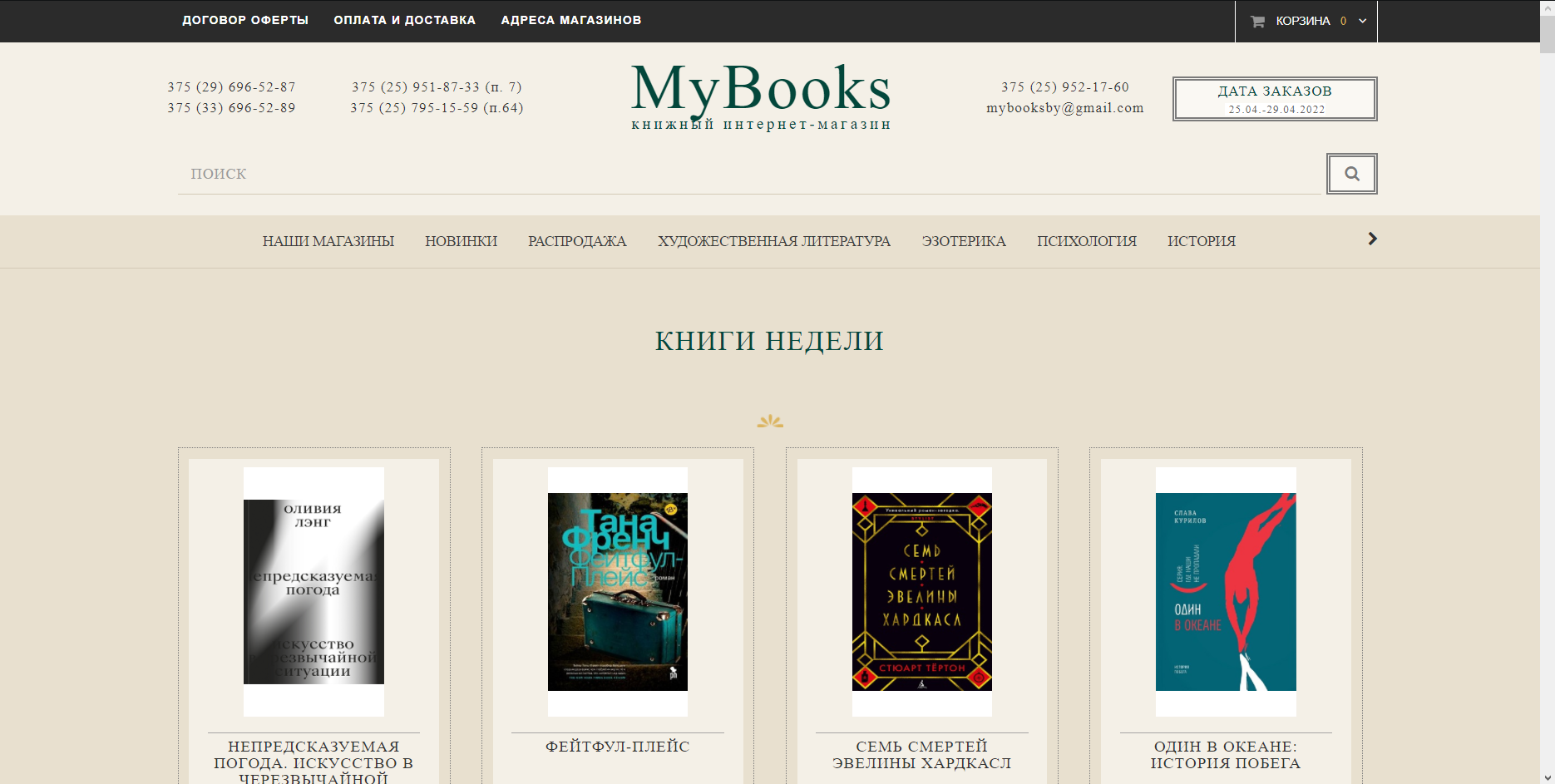


Рисунок 1.3 – Интерфейс веб-приложение myboox.by

На рисунке 1.3 представлен интерфейс веб-приложения myboox.by. Из функций можно выделить: гибкий поиск книг, наличие подборок за период.

Плюсы:

* гибкий поиск книг;
* наличие подборок за период;

Минусы:

* устаревший интерфейс.

## Формирование требований к программному средству

Основной задачей является создание базы данных, которой достаточно для построения программного средства для управления интернет-магазином книг.

**Управление пользователями:**

* регистрация пользователей с использованием email, хранение информации о, книгах, имени, данных для доставки;

**Управление избранными книгами:**

* создание и редактирование списков избранных книг;

**Управление заказами:**

* создание, учет и управление заказом;
* отслеживание статуса заказа (оформлен, доставляется, доставлен, ошибка);

**Управление компенсациями:**

* возможность создания и учета компенсаций с указанием статуса (ожидание, подтверждено, отклонено), суммы, валюты и описания;

**Связи между таблицами:**

* поддержание связей между данными таблиц (например, связь между пользователями и их данными о доставке, избранными книгами);

# Анализ и разработка функциональных требований

## Используемые технологии при разработке

SQL Server Management Studio (SSMS) - это интегрированная среда разработки (IDE), используемая для работы с базами данных Microsoft SQL Server. SSMS включает в себя широкий спектр инструментов и функций, которые делают ее мощным инструментом для разработчиков, администраторов баз данных и других пользователей SQL Server.

Вот некоторые конкретные примеры того, как SSMS может использоваться:

Создание таблиц: SSMS можно использовать для создания таблиц для хранения данных.

Создание представлений: SSMS можно использовать для создания представлений для отображения данных из таблиц в удобном для пользователя формате.

Создание хранимых процедур: SSMS можно использовать для создания хранимых процедур для выполнения повторяющихся задач.

Выполнение запросов: SSMS можно использовать для выполнения запросов к базе данных для получения данных.

Мониторинг производительности: SSMS можно использовать для мониторинга производительности базы данных. Это позволяет администраторам баз данных выявлять и устранять проблемы с производительностью.

Резервное копирование и восстановление: SSMS можно использовать для резервного копирования и восстановления базы данных. Это позволяет защитить данные от потери.

Стандартизация: SSMS является стандартным инструментом для работы с базами данных SQL Server. Это означает, что разработчики и администраторы баз данных, знакомые с SSMS, смогут легко использовать его для работы.

Поддержка различных платформ: SSMS доступен для Windows, macOS и Linux. Это означает, что приложение, разработанное с использованием SSMS, может работать на различных платформах.

Совместимость с другими продуктами Microsoft: SSMS совместим с другими продуктами Microsoft, такими как Visual Studio и SQL Server Integration Services. Это позволяет разработчикам интегрировать приложение с другими приложениями и системами.

## Функциональные требование к программному средству

### Клиент

* просмотр, редактирование и удаление книг из избранного;
* просмотр, редактирование, добавление новых учетных записей.
* авторизация в системе

### Общие требования:

**Интеграция и безопасность:**

* защита данных и обеспечение безопасности доступа к аккаунтам клиентов с использованием шифрования и других мер безопасности;

**Удобство использования и интерфейс:**

* интуитивно понятный пользовательский интерфейс для просмотра книг, добавления в избранное, оформления заказа;
* организация информации таким образом, чтобы пользователи могли легко находить интересующие их книги;

**Масштабируемость и производительность:**

* обеспечение производительности при работе с большим объемом данных и возможность масштабирования приложения для поддержки роста бизнеса.

# Инфологическая модель предметной области

## Описание предметной области

Предметная область базы данных приложения для интернет-магазина книг обширна и выключает в себя разнообразные аспекты, связанные с автоматизацией процессов, учётом и управлением продуктов, а также обеспечением безопасности и конфиденциальности данных.

## Модель данных



Рисунок 3.1 – Инфологическая модель базы данных

База данных состоит из следующих сущностей: compensation\_subtype,

compensation\_type, compensation\_type\_on\_subtype, user\_role, user, user\_invoice\_data, compensation, invoice, file, user, user\_role, pass\_token.

Таблица user служит для хранения пользователей:

* id: идентификатор пользователя;
* slack\_id: идентификатор Slack API;
* chat\_id: идентификатор переписки для Slack API;
* first\_name: имя пользователя;
* last\_name: фамилия пользователя;
* icon\_url: ссылка на аватарку пользователя;
* status: статус пользователя (активен, заблокирован, удалён)

Таблица invoice служит для хранения чеков пользователей:

* id: идентификатор чека;
* user\_id: идентификатор пользователя, которому принадлежит чек;
* created\_at: дата создания чека;
* fix\_gross\_amount: размер заработной платы пользователя;
* currency: валюта ЗП;
* company\_bank\_account\_type: тип банковского счёта компании, с которого будет переводится ЗП для сотрудника;
* invoice\_status: статус чека (создан, отправлен, подтверждён, отклонён, в процессе отправки, отправка не удалась);
* period\_from: дата начала периода выплаты;
* period\_to: дата окончания периода выплаты;
* billable\_workload: оплачиваемая нагрузка;
* comment: комментарий к чеку;
* pdf\_file\_slack\_id: идентификатор PDF файла с чеком отправленного пользователю через Slack API.
* pdf\_file\_slack\_download\_url: ссылка для скачивания файла из Slack.
* unpaid\_leaves: количество неоплачиваемых дней за период.

Таблица compensation предназначена для хранения бонусов и компенсаций пользователей:

* id: идентификатор компенсации;
* invoice\_id: идентификатор чека, к которому принадлежит компенсация;
* compensation\_status: статус компенсации (одобрена, отклонена, в процессе)
* created\_at: дата создания компенсации;
* amount: сумма компенсации;
* currency: валюта компенсации;
* description: описание;
* compensation\_subtype\_id: идентификатор подтипа компенсации;
* compensation\_type\_id: идентификатор типа компенсации;

Таблица compensation\_type предназначена для хранения типов компенсаций:

* id: идентификатор типа компенсации;
* display\_name\_en: имя типа компенсации;

Таблица compensation\_subtype предназначена для хранения типов компенсаций:

* id: идентификатор подтипа компенсации;
* display\_name\_en: имя подтипа компенсации;

Таблица compensation\_type\_on\_subtype отображает связь типа и подтипа компенсации, т.е. к какому типу относится тот или иной подтип:

* compensation\_type\_id: идентификатор типа компенсации;
* compensation\_subtype\_id: идентификатор подтипа компенсации;

Таблица user\_invoice\_data хранит информацию пользователя для создания чека:

* id: идентификатор данных для чека;
* user\_id: идентификатор пользователя, которому принадлежат данные для чека;
* full\_name: полное имя пользователя из документа удостоверяющего личность;
* legal\_address: адрес проживания пользователя;
* iban: номер счёта для зачисления ЗП;
* bank\_name: имя банка, которому принадлежит счёт;
* bank\_address: адрес банка;
* swift: swift номер для перевода;
* email: email пользователя;
* contract\_date: дата заключения контракта;
* description: описание чека;
* invoice\_data\_setup\_complete: флаг отображающий то, полностью ли заполнена информация для чека;

Таблица file хранит файлы с подписями сотрудников и прочие системные файлы:

* id: идентификатор файла;
* user\_id: идентификатор пользователя, которому принадлежит файл;
* file\_type: тип файла (системный, подпись пользователя);
* data: бинарные данные файла;
* properties: свойства или дополнительные параметры для файла;
* file\_name: имя файла;

Таблица user\_role хранит роли пользователей:

* id: идентификатор роли;
* role\_type: тип роли;
* role\_name: название роли;

# Схема базы данных

-- Создание таблицы книг (Books)

CREATE TABLE Books (

book\_id INT PRIMARY KEY,

title NVARCHAR(255),

author\_id INT,

genre\_id INT,

price DECIMAL(10, 2),

publication\_date DATE,

description NVARCHAR(MAX)

);

-- Создание таблицы авторов (Authors)

CREATE TABLE Authors (

author\_id INT PRIMARY KEY,

author\_name NVARCHAR(100)

);

-- Создание таблицы жанров (Genres)

CREATE TABLE Genres (

genre\_id INT PRIMARY KEY,

genre\_name NVARCHAR(100)

);

-- Создание таблицы клиентов (Customers)

CREATE TABLE Customers (

customer\_id INT PRIMARY KEY,

customer\_name NVARCHAR(100),

email NVARCHAR(100),

address NVARCHAR(255)

);

-- Создание таблицы заказов (Orders)

CREATE TABLE Orders (

order\_id INT PRIMARY KEY,

customer\_id INT,

order\_date DATE,

total\_amount DECIMAL(10, 2),

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id)

);

-- Создание таблицы деталей заказов (Order\_Details)

CREATE TABLE Order\_Details (

order\_detail\_id INT PRIMARY KEY,

order\_id INT,

book\_id INT,

quantity INT,

unit\_price DECIMAL(10, 2),

FOREIGN KEY (order\_id) REFERENCES Orders(order\_id),

FOREIGN KEY (book\_id) REFERENCES Books(book\_id)

);

-- Создание таблицы отзывов (Reviews)

CREATE TABLE Reviews (

review\_id INT PRIMARY KEY,

book\_id INT,

customer\_id INT,

rating INT,

comment NVARCHAR(MAX),

FOREIGN KEY (book\_id) REFERENCES Books(book\_id),

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id)

);

-- Создание таблицы списка желаемого (Wishlist)

CREATE TABLE Wishlist (

wishlist\_id INT PRIMARY KEY,

customer\_id INT,

book\_id INT,

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id),

FOREIGN KEY (book\_id) REFERENCES Books(book\_id)

);

-- Создание таблицы корзины (Cart)

CREATE TABLE Cart (

cart\_id INT PRIMARY KEY,

customer\_id INT,

book\_id INT,

quantity INT,

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id),

FOREIGN KEY (book\_id) REFERENCES Books(book\_id)

);

-- Создание таблицы администраторов (Admins)

CREATE TABLE Admins (

admin\_id INT PRIMARY KEY,

admin\_name NVARCHAR(100),

username NVARCHAR(100),

password NVARCHAR(100)

);

-- Создание таблицы деталей доставки (Shipping\_Details)

CREATE TABLE Shipping\_Details (

shipping\_id INT PRIMARY KEY,

order\_id INT,

address NVARCHAR(255),

shipping\_date DATE,

status NVARCHAR(50),

FOREIGN KEY (order\_id) REFERENCES Orders(order\_id)

);

-- Таблица платежей (Payments)

CREATE TABLE Payments (

payment\_id INT PRIMARY KEY,

order\_id INT,

payment\_date DATE,

amount DECIMAL(10, 2),

payment\_status NVARCHAR(50),

FOREIGN KEY (order\_id) REFERENCES Orders(order\_id)

);

-- Таблица скидок (Discounts)

CREATE TABLE Discounts (

discount\_id INT PRIMARY KEY,

discount\_code NVARCHAR(50),

discount\_amount DECIMAL(10, 2),

start\_date DATE,

end\_date DATE

);

-- Таблица категорий книг (BookCategories)

CREATE TABLE BookCategories (

category\_id INT PRIMARY KEY,

category\_name NVARCHAR(100)

);

-- Таблица связи книг и их категорий (Book\_Category\_Link)

CREATE TABLE Book\_Category\_Link (

book\_id INT,

category\_id INT,

PRIMARY KEY (book\_id, category\_id),

FOREIGN KEY (book\_id) REFERENCES Books(book\_id),

FOREIGN KEY (category\_id) REFERENCES BookCategories(category\_id)

);

-- Таблица издательств (Publishers)

CREATE TABLE Publishers (

publisher\_id INT PRIMARY KEY,

publisher\_name NVARCHAR(100),

headquarters\_location NVARCHAR(255)

);

-- Таблица связи книг и их издателей (Book\_Publisher\_Link)

CREATE TABLE Book\_Publisher\_Link (

book\_id INT,

publisher\_id INT,

PRIMARY KEY (book\_id, publisher\_id),

FOREIGN KEY (book\_id) REFERENCES Books(book\_id),

FOREIGN KEY (publisher\_id) REFERENCES Publishers(publisher\_id)

);

-- Таблица авторских прав (Copyrights)

CREATE TABLE Copyrights (

copyright\_id INT PRIMARY KEY,

book\_id INT,

copyright\_year INT,

holder\_name NVARCHAR(100),

FOREIGN KEY (book\_id) REFERENCES Books(book\_id)

);

-- Таблица оценок пользователей (UserRatings)

CREATE TABLE UserRatings (

rating\_id INT PRIMARY KEY,

customer\_id INT,

book\_id INT,

user\_rating DECIMAL(3, 2),

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id),

FOREIGN KEY (book\_id) REFERENCES Books(book\_id)

);

-- Таблица сессий пользователей (UserSessions)

CREATE TABLE UserSessions (

session\_id INT PRIMARY KEY,

customer\_id INT,

login\_time DATETIME,

logout\_time DATETIME,

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id)

);

-- Таблица логов активности (ActivityLogs)

CREATE TABLE ActivityLogs (

log\_id INT PRIMARY KEY,

customer\_id INT,

activity\_description NVARCHAR(MAX),

activity\_time DATETIME,

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id)

);

-- Таблица рекомендаций (Recommendations)

CREATE TABLE Recommendations (

recommendation\_id INT PRIMARY KEY,

book\_id INT,

recommended\_book\_id INT,

FOREIGN KEY (book\_id) REFERENCES Books(book\_id),

FOREIGN KEY (recommended\_book\_id) REFERENCES Books(book\_id)

);

-- Таблица событий корзины (CartEvents)

CREATE TABLE CartEvents (

event\_id INT PRIMARY KEY,

customer\_id INT,

book\_id INT,

event\_type NVARCHAR(50),

event\_time DATETIME,

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id),

FOREIGN KEY (book\_id) REFERENCES Books(book\_id)

);

-- Таблица отчетов о продажах (SalesReports)

CREATE TABLE SalesReports (

report\_id INT PRIMARY KEY,

report\_name NVARCHAR(100),

report\_date DATE,

sales\_amount DECIMAL(15, 2)

);

-- Добавление внешних ключей для Books

ALTER TABLE Books

ADD CONSTRAINT FK\_Books\_Authors FOREIGN KEY (author\_id) REFERENCES Authors(author\_id);

-- Добавление внешних ключей для Books

ALTER TABLE Books

ADD CONSTRAINT FK\_Books\_Genres FOREIGN KEY (genre\_id) REFERENCES Genres(genre\_id);

-- Добавление внешних ключей для SalesReports

ALTER TABLE SalesReports

ADD CONSTRAINT FK\_SalesReports\_Books FOREIGN KEY (report\_id) REFERENCES Books(book\_id);

-- Добавление внешних ключей для Genres

ALTER TABLE Genres

ADD CONSTRAINT FK\_Genres\_Books FOREIGN KEY (genre\_id) REFERENCES Books(book\_id);

-- Добавление внешних ключей для Discounts

ALTER TABLE Discounts

ADD CONSTRAINT FK\_Discounts\_Books FOREIGN KEY (discount\_id) REFERENCES Books(book\_id);

-- Добавление внешних ключей для Authors

ALTER TABLE Authors

ADD CONSTRAINT FK\_Authors\_Books FOREIGN KEY (author\_id) REFERENCES Books(book\_id);

-- Добавление внешних ключей для Admins

ALTER TABLE Admins

ADD CONSTRAINT FK\_Admins\_Books FOREIGN KEY (admin\_id) REFERENCES Books(book\_id);

-- Триггеры

-- Триггер для проверки аномалий даты заказа

CREATE TRIGGER CheckOrderDate

ON Orders

AFTER INSERT

AS

BEGIN

IF EXISTS (

SELECT 1

FROM inserted

WHERE order\_date > GETDATE() -- Условие для проверки даты

)

BEGIN

RAISERROR ('Недопустимая дата заказа.', 16, 1)

ROLLBACK TRANSACTION

END

END;

-- Триггер для проверки уникальности покупателя по email адресу

CREATE TRIGGER CheckCustomerEmail

ON Customers

AFTER INSERT, UPDATE

AS

BEGIN

IF EXISTS (

SELECT email, COUNT(\*)

FROM inserted

GROUP BY email

HAVING COUNT(\*) > 1 -- Проверка уникальности email адреса

)

BEGIN

RAISERROR ('Покупатель с таким email уже существует.', 16, 1)

ROLLBACK TRANSACTION

END

END;

-- Триггер для каскадного удаления книг указанного автора при удалении автора

CREATE TRIGGER CascadeDeleteAuthorBooks

ON Authors

AFTER DELETE

AS

BEGIN

DELETE FROM Books WHERE author\_id IN (SELECT deleted.author\_id FROM deleted)

END;

-- Триггер для запрета добавления одной и той же книги в избранное несколько раз

CREATE TRIGGER PreventDuplicateWishlist

ON Wishlist

AFTER INSERT

AS

BEGIN

IF EXISTS (

SELECT book\_id, customer\_id, COUNT(\*)

FROM inserted

GROUP BY book\_id, customer\_id

HAVING COUNT(\*) > 1 -- Проверка на дубликаты в избранном

)

BEGIN

RAISERROR ('Книга уже есть в избранном.', 16, 1)

ROLLBACK TRANSACTION

END

END;

-- Представления

-- Представление для получения книг и их авторов

CREATE VIEW BooksAndAuthors AS

SELECT b.book\_id, b.title, b.price, b.publication\_date, b.description, a.author\_name

FROM Books b

JOIN Authors a ON b.author\_id = a.author\_id;

-- Представление для получения информации о пользователях и их истории заказов

CREATE VIEW CustomerOrderHistory AS

SELECT c.customer\_id, c.customer\_name, c.email, o.order\_id, o.order\_date, o.total\_amount

FROM Customers c

LEFT JOIN Orders o ON c.customer\_id = o.customer\_id;

-- Представление для получения информации об издательствах и их книгах

CREATE VIEW PublishersAndBooks AS

SELECT p.publisher\_id, p.publisher\_name, p.headquarters\_location, b.title

FROM Publishers p

LEFT JOIN Book\_Publisher\_Link bpl ON p.publisher\_id = bpl.publisher\_id

LEFT JOIN Books b ON bpl.book\_id = b.book\_id;

# Подробное описание бизнес-логики

## Триггеры

Триггер для проверки аномалий даты заказа;

Триггер для проверки уникальности покупателя по email адресу;

Триггер для каскадного удаления книг указанного автора при удалении автора;

Триггер для запрета добавления одной и той же книги в избранное несколько раз.

## Представления

Представление для получения книг их авторов;

Представление для получения информации о пользователях и их истории заказов;

Представление для получения информации об издательствах и их книгах.

# Тестирование и проверка работоспособности программного средства

В ходе тестирования были рассмотрены в действие весь основной заложенный функционал базы данных.

## Триггеры

### Обновление периода чека на период превышающий один месяц

Таблица 1 – Обновление периода чека на период превышающий один месяц

|  |  |
| --- | --- |
| Составляющие теста | Данные и результаты теста |
| Тестовая ситуация | Обновление периода чека на период превышающий один месяц |
| Исходный набор данных: | period\_from: 2023-11-01 00:00:00.000  period\_to: 2023-12-15 00:00:00.000 |
| Ожидаемый результат: | Ошибка: Invalid period value |
| Полученный результат: |  |

### Добавление компенсации с валютой не совпадающей с валютой чека

Таблица 2 – Добавление компенсации с валютой не совпадающей с валютой чека

|  |  |
| --- | --- |
| Составляющие теста | Данные и результаты теста |
| Тестовая ситуация | Добавление компенсации с валютой не совпадающей с валютой чека |
| Исходный набор данных: | invoice\_id: 3f232ce5-d465-40a7-90f0-aa19aa8a82df,  compensation\_status: PENDING,  created\_at: CURRENT\_TIMESTAMP,  amount: 123.23,  currency: EUR,  description: null,  compensation\_subtype\_id: 12,  compensation\_type\_id: 1 |
| Ожидаемый результат: | Ошибка: Compensation and invoice currency have to be the same |
| Полученный результат: |  |

### Добавление файла с подписью

Таблица 3 – Добавление файла с подписью

|  |  |
| --- | --- |
| Составляющие теста | Данные и результаты теста |
| Тестовая ситуация | Добавление файла с подписью |
| Исходный набор данных: | user\_id: 3ab23575-f4b3-4586-9a9e-14029b10bed8,  file\_type: USER\_SIGNATURE  data: <Binary Data>  properties: null  file\_name: null |
| Ожидаемый результат: | user\_invoice\_data.invoice\_data\_setup\_complete = true |
| Полученный результат: |  |

### Обновление даты начала периода чека

Таблица 4 – Обновление даты начала периода чека

|  |  |
| --- | --- |
| Составляющие теста | Данные и результаты теста |
| Тестовая ситуация | Обновление даты начала периода чека, чтобы дата начала была позже даты окончания |
| Исходный набор данных: | period\_from: 2023-11-01 00:00:00.000  period\_to: 2023-10-01 00:00:00.000 |
| Ожидаемый результат: | Ошибка: The end date of the period should be later than the start date. |
| Полученный результат: |  |

### Обновление количества неоплачиваемых дней

Таблица 5 – Обновление количества неоплачиваемых дней

|  |  |
| --- | --- |
| Составляющие теста | Данные и результаты теста |
| Тестовая ситуация | Обновление количества неоплачиваемых дней так, чтобы это количество было больше количества дней в месяце |
| Исходный набор данных: | unpaid\_leaves: 50 |
| Ожидаемый результат: | Ошибка: The number of unpaid leave days cannot exceed the number of days between period\_from and period\_to. |
| Полученный результат: |  |

## Представления

### Получение чеков, компенсаций и сотрудников, которым принадлежат эти чеки.

Таблица 6 – Получение чеков, компенсаций и сотрудников, которым принадлежат эти чеки.

|  |  |
| --- | --- |
| Составляющие теста | Получение чеков, компенсаций и сотрудников, которым принадлежат эти чеки. |
| Тестовая ситуация | Получение чеков, компенсаций и сотрудников, которым принадлежат эти чеки. |
| Исходный набор данных: | - |
| Ожидаемый результат: | Список из одного чека |
| Полученный результат: |  |

### Получение чеков, компенсаций, типов и подтипов компенсаций.

Таблица 7 – Получение чеков, компенсаций, типов и подтипов компенсаций.

|  |  |
| --- | --- |
| Составляющие теста | Получение чеков, компенсаций, типов и подтипов компенсаций. |
| Тестовая ситуация | Получение чеков, компенсаций, типов и подтипов компенсаций. |
| Исходный набор данных: | - |
| Ожидаемый результат: | Список из одного чека |
| Полученный результат: |  |

Заключение

Результатом работы над курсовым проектом стало создание базы данных для исправного и рабочего прототипа книжного интернет-магазина. Ожидается, что полученный прототип получит дальнейшее развитие и уже сможет задействовать в себе полный перечень таблиц из разработанной модели БД, тем самым реализовав до конца весь полный перечень технических требований.

Были детально изучены заданная предметная область, основные направления e-commerce, а также подходы к открытию собственного интернет-магазина, тонкости книгопродажи и букинистики, а также теория реляционных баз данных, архитектуры клиент-серверного приложения.

Было проведено ознакомление с интерфейсом и примерами использования Oracle SQL Developer, это средство было применено при разработке базы данных вкупе с SQLServerManagmentStudio (моделирование).

Были тщательно продуманы и разработаны архитектура приложения, оперируемые в ней данные, а также все необходимые алгоритмы для получения требуемого результата.

Полученное приложение было разработано и многократно протестировано с соблюдением всех принципов разработки программного обеспечения.

В ходе проекта были выявлены принципиальные моменты проектирования базы данных и ее влияние на работу приложения. Опыт разработки подчеркнул важность обеспечения высокой степени надежности системы и ее безопасности при обработке финансовых данных.

В дальнейшем, проект может быть расширен добавлением клиент-серверного приложения, добавлением дополнительных функций, таких как проверка наличия книг в физических магазинах.

Список использованной литературы

[1] Национальный статистический комитет Республики Беларусь [Электронный ресурс]. — Режим доступа: https://belstat.by/. — Дата доступа: 05.10.2023.

[2] Юрасов, А. В. Электронная коммерция: учеб. пособие / А. В. Юрасов. — М.: Дело, 2003. — 480 с.

[3] Куликов, С. С. Реляционные базы данных в примерах / С. С. Куликов. — EPAM Systems, RD Dep., 2021. — 422 с.

[4] Интернет-магазин Oz.by [Электронный ресурс]. — Режим доступа: https://oz.by/books. — Дата доступа: 13.10.2023.

[5] Интернет-магазин Ozon.ru [Электронный ресурс]. — Режим доступа: https://ozon.ru/. — Дата доступа: 14.10.2023.

[6] Интернет-магазин MyBooks.by [Электронный ресурс]. — Режим доступа: https://mybooks.by/. — Дата доступа: 15.10.2023.

[7] Интернет-магазин Book Depository [Электронный ресурс]. — Режим доступа: <https://www.bookdepository.com/>. — Дата доступа: 17.10.2023

[8] MySQL Documentation [Электронный ресурс]. – Режим доступа: <https://dev.mysql.com/doc/>. Дата доступа: 01.10.2023.

[9] MySQL Workbench [Электронный ресурс]. – Режим доступа: <https://www.mysql.com/products/workbench/>. Дата доступа: 05.09.2023.

[10] SQL tutorials [Электронный ресурс]. – Режим доступа: <https://www.w3schools.com/sql/>. Дата доступа: 06.09.2023.

[11] Куликов, Работа с MySQL, MS SQL Server и Oracle (2-е издание) / Куликов С.С. – М. Четыре четверти, 2021.

ВЕДОМОСТЬ ДОКУМЕНТОВ

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Обозначение | | | | | Наименование | | | | Дополнительные сведения | |
|  | | | | | Текстовые документы | | | |  | |
|  | | | | |  | | | |  | |
| БГУИР КП 1–40 01 01 05 СП | | | | | Пояснительная записка | | | | 31 с. | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | | Графические документы | | | |  | |
|  | | | | |  | | | |  | |
| ГУИР 051003-01 СА | | | | | База данных интернет-магазина книг | | | | Формат А1 | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  |  |  |  |  | БГУИР КП I- 40 01 01 05 ПЗ | | | | | |
|  |  |  |  |  |
|  |  |  |  |  |
| Изм | Лист | № докум. | Подп. | Дата | База данных интернет-магазина книг | Литера | | | Лист | Листов |
| Разраб. | | Гаркушенко С.С. |  |  | Т |  |  | 31 | 31 |
| Провер. | | Колотыгин К.Е. |  |  | Кафедра ПОИТ  гр. 051003 | | | | |
|  | |  |  |  |
|  | |  |  |  |
|  | |  |  |  |