МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ   
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра информационных систем и технологий

Специальность 1-40 05 01 «Информационные системы и технологии»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА КУРСОВОГО ПРОЕКТА**

по дисциплине «Базы данных»

Тема: «База данных интернет-магазина велосипедов с применением Full Text Search-технологий СУБД»

**Исполнитель**

студент 2 курса 2 группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.О. Лемешевский

подпись, дата

**Руководитель**

\_\_\_\_\_\_\_\_\_\_\_\_ Н.И. Уласевич

(должность, уч. звание) (подпись, дата)

Допущен(а) к защите \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

дата, подпись

Курсовой проект защищен с оценкой

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.И. Уласевич

подпись дата инициалы и фамилия

Минск 2024

Содержание

[Введение 4](#_Toc161798117)

[1 Аналитический обзор литературы по теме проекта 5](#_Toc161798118)

[1.1 Постановка задачи 5](#_Toc161798119)

[1.1.1 Аналог iBike 5](#_Toc161798120)

[1.1.2 Аналог Source BMX 6](#_Toc161798121)

[1.2 Функциональные требования 7](#_Toc161798122)

[1.3. Определение вариантов использования 8](#_Toc161798123)

[1.4 Вывод 10](#_Toc161798124)

[2 Анализ и проектирование модели базы данных. 11](#_Toc161798125)

[2.1 UML схема базы данных 11](#_Toc161798126)

[2.2 Описание информационных объектов и ограничений целостности 12](#_Toc161798127)

[2.3 Вывод 13](#_Toc161798128)



# Введение

Цель данной работы заключается в создании реляционной базы данных для интернет-магазина , которая обеспечивает пользователю доступ к имеющимся в нем товарам. В рамках работы также необходимо применить соответствующую технологию, Full Text Search, которая будет демонстрировать определенную функциональность базы данных, включая полнортекствый поиск товаров по индексу, предложению и т.д.

База данных - это совокупность взаимосвязанных данных, которые обычно хранятся в электронном виде в компьютерной системе. БД используются для хранения, организации и управления большим объемом структурированных и неструктурированных данных. Реляционная база данных является наиболее распространенной формой организации данных, в которой данные представлены в виде таблиц, состоящих из строк и столбцов, где каждый столбец представляет атрибут, а каждая строка представляет кортеж или запись. В данной работе для управления базой данных была выбрана СУБД «Oracle», поскольку эта система обладает высокой надежностью и производительностью, что позволяет обеспечить эффективное хранение, обработку и управление данными.

Для обеспечения функциональности приложения используются технология полнотекстового поиска, которая позволяет приложениям и пользователям выполнять полнотекстовые запросы к символьным данным в таблицах.

В пояснительной записке содержится информация о сопоставимых продуктах, структуре и реализации проекта, а также инструкции по использованию приложения.

1. Аналитический обзор литературы по теме проекта

## Постановка задачи

Интернет-магазины стали неотъемлемой частью нашей современной жизни. Они предоставляют нам доступ к бесчисленному количеству товаров и услуг прямо из домашних условий. Поэтому, для реализации базы данных интернет-магазина необходимо рассмотреть и объективно оценить возможный функционал базы данных, применимые технологии и т.п. По результатам анализа необходимо составить определенный список функциональных требований к базе данных и диаграмму вариантов использования. Задача проекта: разработать архитектуру приложения, взаимодействие с которой будет понятно любому пользователю. Построить базу данных и выполнить тестирование готового продукта.

Основные требования к реализации:

* Определение ролей (Покупатель, продавец, менеджер);
* Возможность добавления и удаления товара (Продавец, менеджер);
* Возможность оформление заказа товара (Покупатель);
* Поиск товара по категориям, критериям, названию (Покупатель);
* Возможность оценки товара (Покупатель);
* Возможность добавление товара в корзину/избранное (Покупатель);
* Взаимодействие с базой данных при помощи хранимых процедур и функций.

### 1.1.1 Аналог iBike

iBike [1] - это крупнейший экстрим-магазин в Беларуси. Они специализируются на продаже велосипедов BMX и MTB, запчастей, защиты и аксессуаров. Их магазин является официальным представителем многих брендов БМХ, таких как WTP, Eclat, Odyssey, Fuse, BSD, Cult и других. Пример интерфейса данного сервиса представлен на рисунке 1.1.

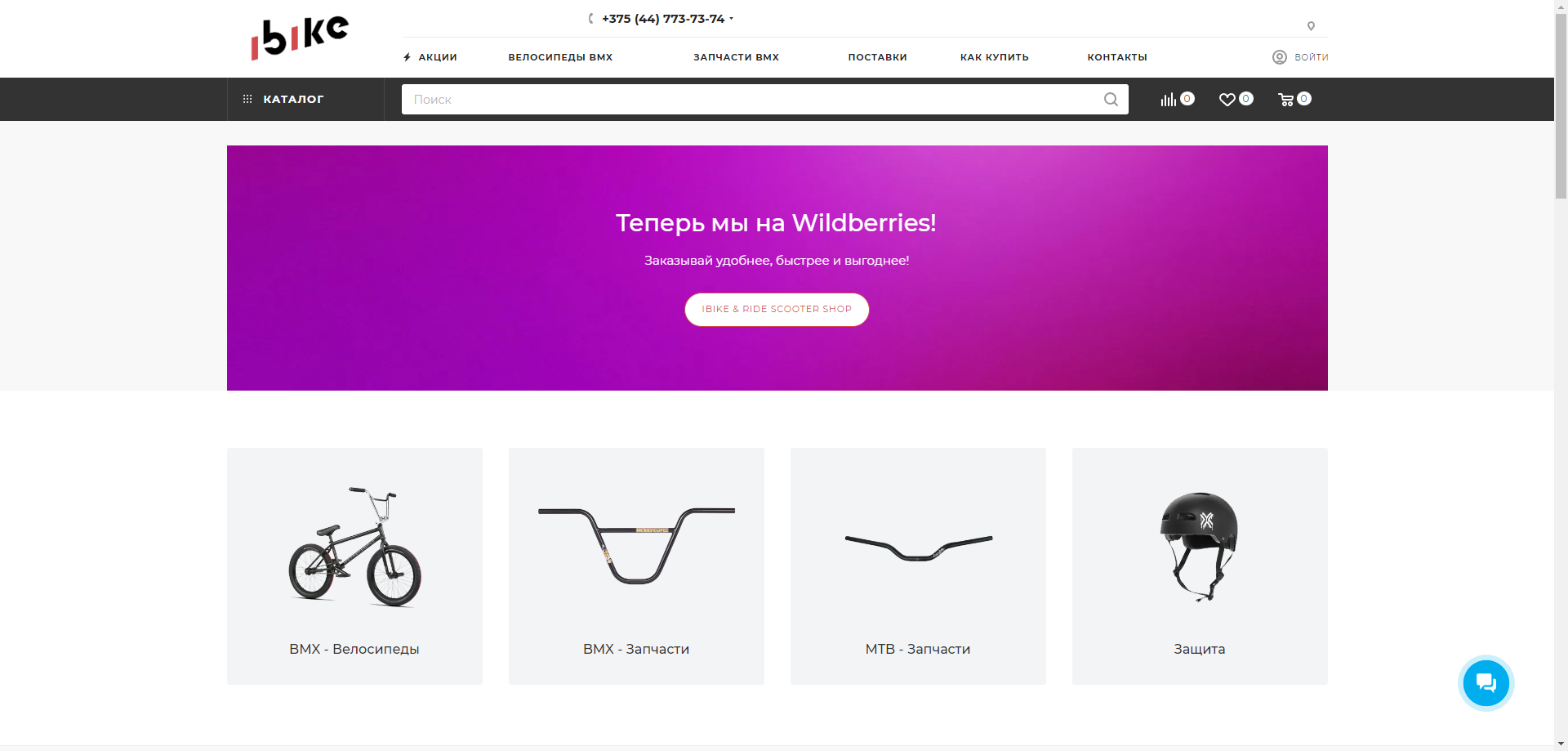


Рисунок 1.1 – Интерфейс сервиса iBike

Одной из основных функций iBike.by является продажа экстрим-велосипедов. Пользователи могут искать велосипеды по бренду, модели, типу или цене. Кроме того, iBike.by предлагает персонализированные рекомендации в соответствии с предпочтениями пользователя, а также список товаров, рекомендованных другими пользователями и кураторами платформы. Пользователи могут создавать свои собственные списки желаемых товаров и делиться ими с друзьями или сохранять товары для последующего просмотра. iBike.by также позволяет пользователям просматривать информацию о товарах, включая оригинальный контент, созданный самой платформой. Другая функция iBike.by - это возможность использовать платформу как социальную сеть, где пользователи могут подписываться на друг друга, просматривать их списки желаемых товаров и рекомендации, а также обмениваться сообщениями. iBike.by также предлагает два варианта покупки: онлайн и в магазине. Онлайн-покупка позволяет получить доступ к полному ассортименту товаров, в то время как покупка в магазине позволяет получить профессиональную консультацию и другие функции.iBike.by использует несколько баз данных в своей архитектуре, включая системы управления базами данных для хранения информации о товарах, включая информацию о брендах, моделях, типах и т.д. Эти системы также используются для хранения информации о пользовательских аккаунтах и настройках. В общем, iBike.by использует масштабируемые и высокопроизводительные базы данных для обеспечения быстрого доступа к большому объему информации о товарах и обработки данных для предоставления персонализированных рекомендаций и других функций платформы.

### 1.1.2 Аналог Source BMX

Source BMX [2] - это популярная платформа для продажи BMX велосипедов, которая позволяет пользователям просматривать и покупать товары онлайн. Source BMX позволяет пользователям просматривать товары на различных устройствах, а также создавать собственные списки желаемых товаров. Source BMX предлагает пользователям доступ к широкому ассортименту BMX велосипедов, запчастей и аксессуаров. Сервис также предлагает персонализированные рекомендации, основанные на предпочтениях пользователя.

Интерфейс Source BMX схож с интерфейсом iBike и предлагает удобный поиск по брендам, моделям и типам товаров. Также сервис позволяет сохранять товары для последующего просмотра. Пример интерфейса данного сервиса представлен на рисунке 1.2

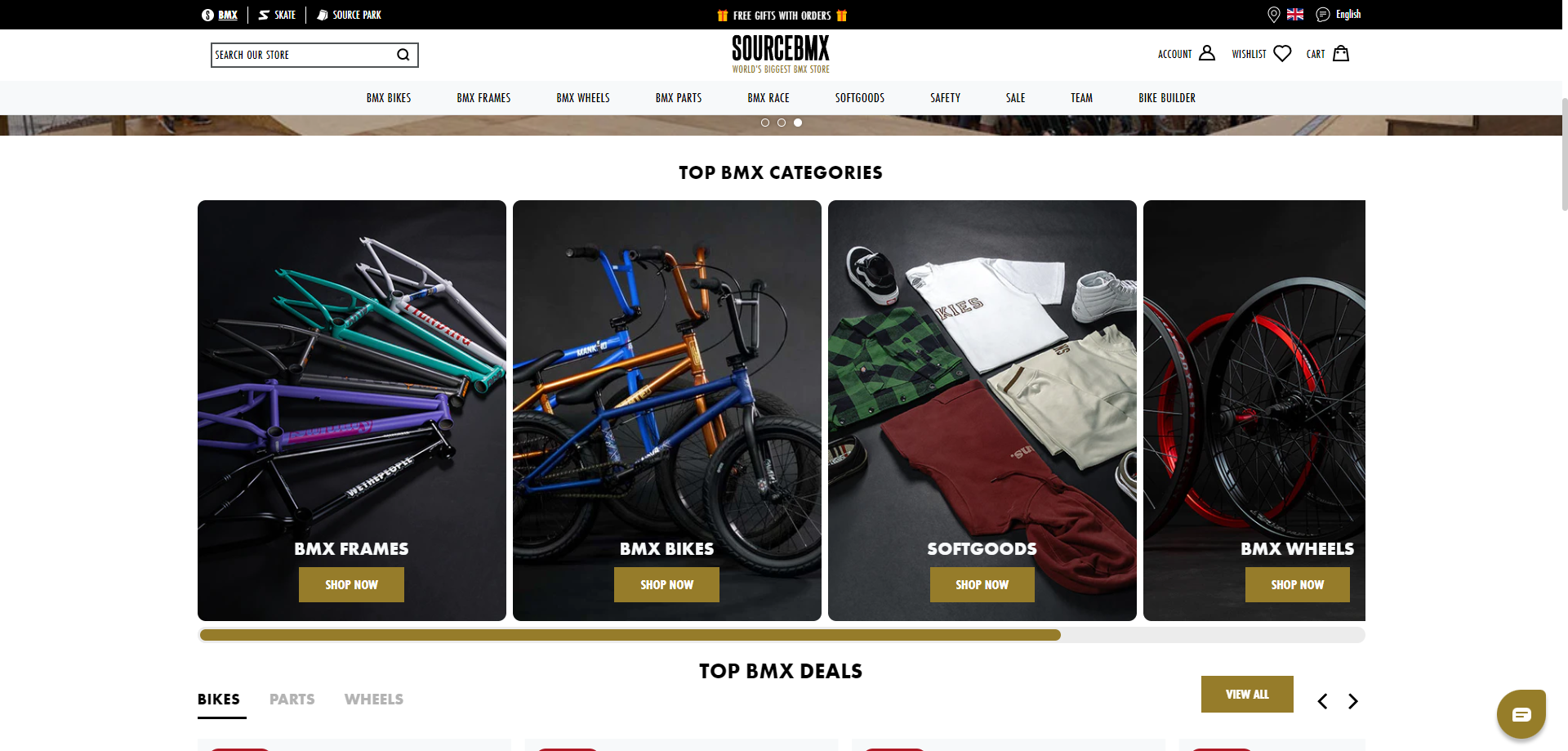


Рисунок 1.2 – Интерфейс сервиса Source BMX

Source BMX может использовать несколько баз данных в своей архитектуре. Возможно, они используют реляционную базу данных, такую как MySQL, для хранения информации о пользователях, такой как логины и пароли, а также информации о покупках и оплате. Для хранения метаданных товаров, включая информацию о брендах, моделях, типах и т.д., а также для предоставления быстрого доступа к этой информации пользователям, они могут использовать NoSQL базу данных, такую как Cassandra. Кроме того, Source BMX может использовать облачную инфраструктуру, например, Amazon Web Services или Google Cloud, для обеспечения высокой доступности и масштабируемости своей платформы.

В целом, Source BMX, вероятно, также использует масштабируемые и высокопроизводительные базы данных для обеспечения быстрого доступа к большому объему информации о товарах и управления информацией о пользователях и их покупках. Однако, конкретные детали об архитектуре и используемых технологиях могут быть доступны только у команды Source BMX

## 1.2 Функциональные требования

Функциональные требования базы данных определяют, как база данных должна обрабатывать данные и предоставлять пользователю определенной роли необходимую функциональность. Это может включать в себя описание того, как данные должны храниться и организовываться, как происходит поиск и выборка данных, каким образом обновляются данные и какие механизмы используются для защиты данных. Кроме того, функциональные требования могут определять интеграцию базы данных с другими системами и программами. Например, для интернет-магазина велосипедов функциональные требования могут включать в себя функции для хранения информации о описании товаров, изображений, поиска товаров по категориям и критериям, создание и удаление карточек товаров, добавление и удаление товаров из корзины, а также функции для оценки товаров.

По результатам рассмотренных аналогов основные задачи для базы данных интернет-магазина:

* разработать процедуры и функции для обработки информации и действий пользователя;
* разработать функционал добавления и удаления товаров из корзины;
* разработать структурную модель базы данных;
* разработать функционал добавления информации о товаре конкретным пользователем;
* разработать функционал оценки товара;
* разработать функционал для заказа товара;
* разработать функционал редактирования/удаления пользователя/товара.
* разработать функционал полнотекстового поиска товаров.
* оформить функционал сохранения определенной мультимедийной информации о товаре (фото).

## 1.3. Определение вариантов использования

Помимо функциональных требований, важно также определить роли пользователей и их варианты использования системы. Варианты использования описывают, как пользователи будут взаимодействовать с системой в зависимости от своих ролей. Варианты использования обычно представляются в виде UML диаграмм, которые позволяют наглядно отобразить взаимодействие между пользователями и системой.

В зависимости от роли пользователя, он может иметь доступ к различным функциям системы. В данном проекте роли пользователей будут следующими:

* Гость
* Покупатель
* Продавец
* Менеджер

На основе предоставленного списка ролей необходимо построить варианты использования. Варианты использования изображены на рисунке 1.3.

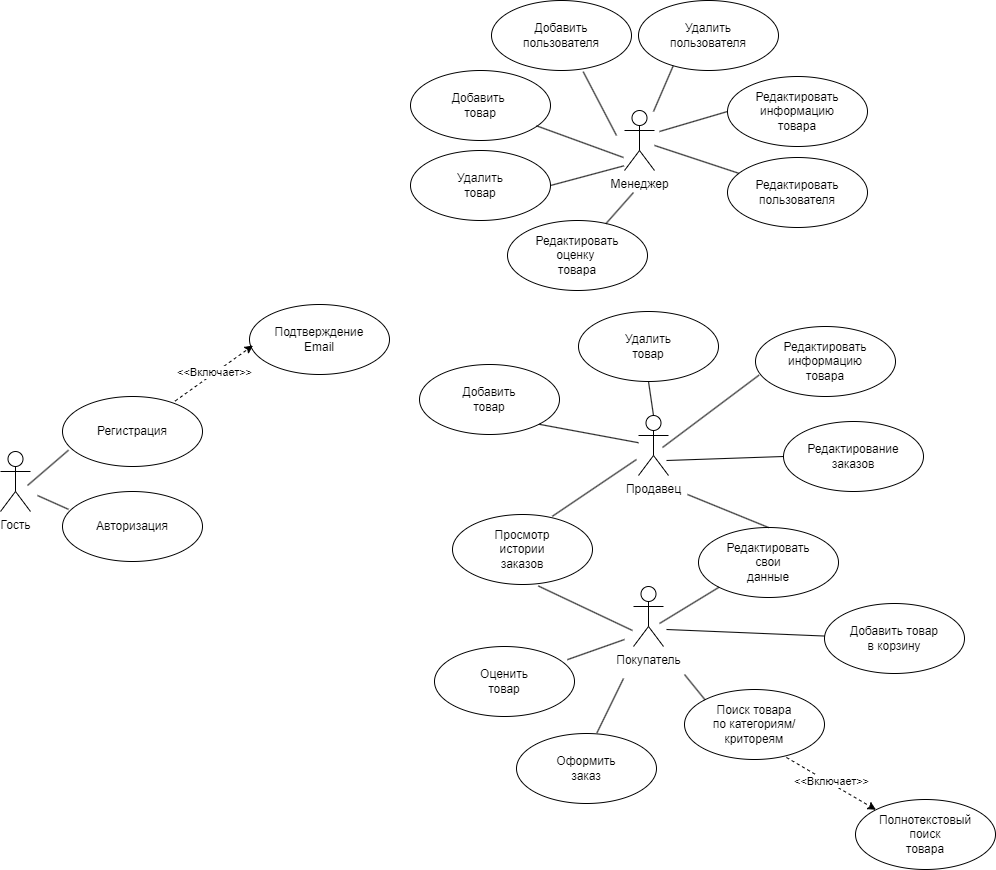


Рисунок 1.3 – UML диаграмма вариантов использования

В начале работы с приложением пользователь является гостем. Ему будет доступна только просмотр всей информации без возможности добавления товара в корзину. После регистрации пользователь становится зарегистрированным пользователем который может быть как покупать (Покупатель), так и продавать (Продавец) товар.

Роль Покупатель получает возможность добавлять понравившиеся товары в корзину, оценивать товары ставя им оценку и оставлять свой комментарий о товаре, осуществлять поиск товара по категориям или критериям, а также пользоваться технологией полнотекстового поиска.

Роль Продавец получает возможность добавлять товары на продажу, изменять параметры продаваемого товара, добавляя к нему описание или медиа контент, он так же может и снимать собственные товары с продажи, просматривать историю заказов, узнав информацию о тех кто заказал товар, Продавец так же может управлять текущими заказами закрывая их или редактируя. Так же помимо основных возможностей пользователю с ролью Продавец ничего не мешает использовать возможности Покупателя.

Оба пользователя с ролью Покупателя и Продавца могут изменять собственные данные: имя, адрес, номер телефона и т.д.

Роль Менеджер заключается в управлении площадкой и мониторинге действий пользователей. Менеджер в качестве администратора имеет доступ к просмотру информации о всех пользователях, в том числе их действиях на площадке. При необходимости Менеджер может редактировать информацию о товарах и пользователях, например, изменять категории и форматировать отзывы у товаров, а также изменять личную информацию о пользователях.

## 1.4 Вывод

Итого, был проведен аналитический обзор аналогов интернет-магазинов, которые уже существуют на рынке. Этот обзор позволил определить основные характеристики и функциональные возможности, которые необходимо предусмотреть в разрабатываемой базе данных. Также были определены функциональные требования базы данных, а также роли пользователей и варианты использования в зависимости от этих ролей. Была разработана UML-диаграмма, на которой отображены основные функции, которые доступны для каждой из ролей пользователей.

1. Анализ и проектирование модели базы данных.

## --2.1 UML схема базы данных

Схема базы данных представляет собой логическую конфигурацию либо целой реляционной базы данных, либо ее части. Схема может существовать как в виде наглядного представления базы данных, так и в виде набора формул (также именуемых «условиями целостности»), которые регулируют ее устройство. Эти формулы выражаются с помощью языка описания данных, например, SQL. Будучи частью словаря данных, схема показывает, как связаны между собой сущности, из которых состоит база данных (таблицы, представления, хранимые процедуры и так далее). Схема базы данных будет представлена на рисунке 2.1.

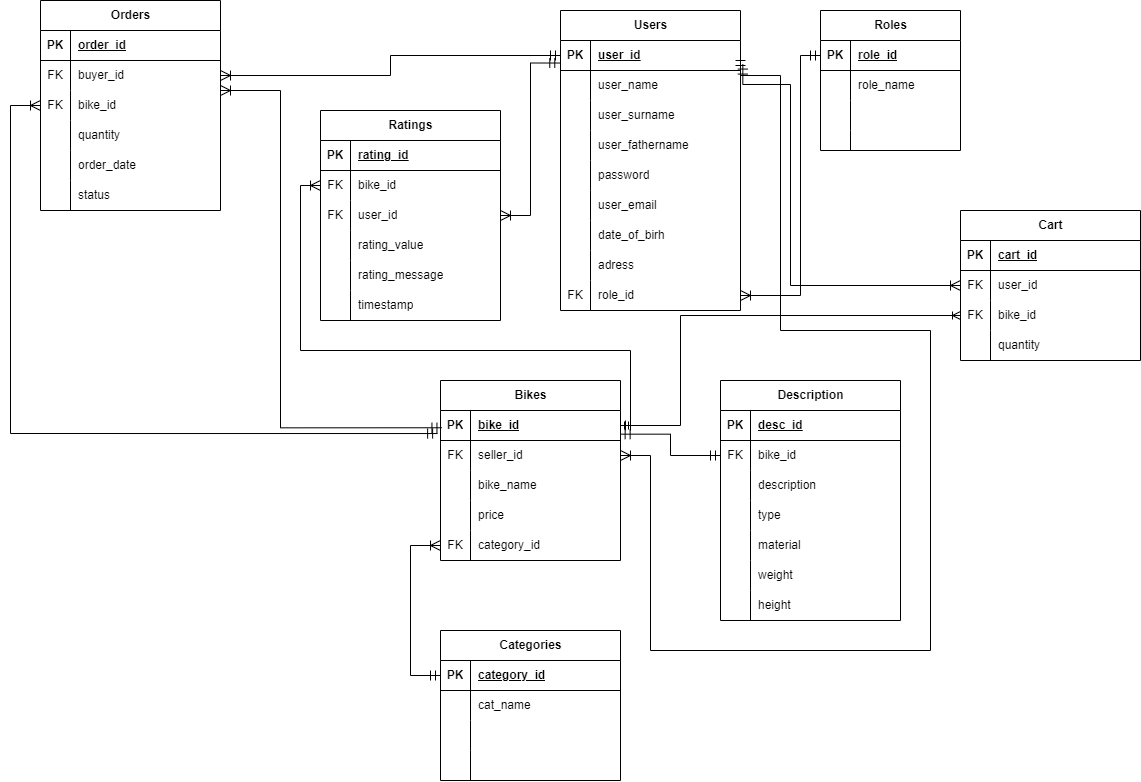
SE

Рисунок 2.1 –Схема базы данных

Таким образом, схема показывает связи между таблицами и полями, а также типы отношений между ними, такие как связи "один-ко-многим" и "многие-ко-многим". Например, таблица Users связана с таблицами Bikes и Orders через внешние ключи seller\_id и buyer\_id. Также видно, что таблица Bikes связана с таблицей Categories и Descriptions через внешний ключ bike\_id. Кроме того, таблица Orders связана с таблицей Bikes по bike\_id, а таблица Ratings связана с товаром через внешний ключ bike\_id и с таблицей Users через внешний ключ user\_id.

## 2.2 Описание информационных объектов и ограничений целостности

Для реализации базы данных было разработано 8 таблиц. В структуру схемы базы данных для проекта входят следующие таблицы: Roles, Users, Ratings, Bikes, Cart, Categories, Description, Orders. Ниже будет описание про каждую из них более подробно. Описание представлено в таблицу 2.1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Roles | role\_id | PK | Идентификатор роли | NUMBER |
| role\_name |  | Текстовое название роли | VARCHAR(255) |
| Users | user\_id | PK | Идентификатор пользователя | NUMBER |
| user\_name |  | Имя пользователя | VARCHAR(255) |
| user\_surname |  | Фамилия пользователя |
| user\_fathername |  | Отчество пользователя |
| user\_email |  | Почта пользователя |
| password |  | Пароль |
| date\_of\_birth |  | Дата рождения |
| address |  | Адрес проживания |
| role\_id | FK | Идентификатор роли | NUMBER |
| Bikes | bike\_id | PK | Идентификатор товара | NUMBER |
| seller\_id | FK | Идентификатор продавца |
| bike\_name |  | Название товара | VARCHAR(255) |
| price |  | Цена товара | DECIMAL(10,2) |
| category\_id | FK | Категория товара | NUMBER |
| Description | desc\_id | PK | Идентификатор описания | NUMBER |
| bike\_id | FK | Идентификатор товара |
| description |  | Словесное описание товара | VARCHAR(255) |
| type |  | Тип товара | VARCHAR(255) |
| material |  | Материл изготовления | CHAR(50) |
| weight |  | Масса товара | FLOAT(5) |
| height |  | Высота | FLOAT(5) |
| Categories | сategory\_id | PK | Идентификатор категории товара | NUMBER |
| сat\_name |  | Название категории товара | VARCHAR(255) |
| Orders | order\_id | PK | Идентификатор заказа | NUMBER |
| buyer\_id | FK | Идентификатор покупателя |
| bike\_id | FK | Идентификатор товара |
| quantity |  | Количество | FLOAT(5) |
| order\_date |  | Дата заказа | DATETIME |
| status |  | Статус заказа | VARCHAR(255) |
| Ratings | rating\_id | PK | Идентификатор оценки товара | NUMBER |
| bike\_id | FK | Идентификатор товара |
| user\_id | FK | Идентификатор пользователя |
| rating\_value |  | Значение оценки |
| rating\_message |  | Комментарий к оценке | VARCHAR(255) |
| timestamp |  | Дата оценки | DATETIME |
| Cart | cart\_id | PK | Идентификатор корзины | NUMBER |
| user\_id | FK | Идентификатор пользователя |
| bike\_id | FK | Идентификатор товара |
| quantity |  | Количество |

Таблица 2.1 − Описание ограничений целостности

## 2.3 Вывод

Анализ и проектирование модели базы данных представляют собой важный этап разработки системы. Была разработана UML-схема базы данных, которая является логической конфигурацией реляционной базы данных. Схема показывает связи между таблицами и полями, а также типы отношений между ними. Для реализации базы данных было разработано восемь таблиц: Roles, Users, Ratings, Bikes, Cart, Categories, Description, Orders. Каждая из них имеет свою специфику и связана с другими таблицами через внешние ключи.