Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет   
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина «Технологии обработки финансовой документации»

|  |  |
| --- | --- |
|  |  |

**РАЗРАБОТКА БАЗЫ ДАННЫХ**

для проекта

«Веб-приложения для размещения товаров и услуг»

Выполнили Д. С. Кончик, гр. 153503

Е. С. Кахновский, гр. 153503

Проверил В. Д. Иванович

Минск 2024

1. ER-ДИАГРАММА

Для разработки базы данных веб-приложения для размещения товаров и услуг сначала необходимо выделить все основные сущности проекта и определить взаимосвязи между ними. На основе этого можно построить ER-модель, которая наглядно отразит структуру данных и покажет, как данные будут взаимодействовать в системе.

В ER-модели выделяются три основные типа связей между сущностями:

1. «Один-к-одному» – тип связи, в котором один экземпляр сущности связан только с одним экземпляром другой сущности. Например, каждый пользователь может иметь только одну учетную запись с уникальными настройками и личной информацией.
2. «Один-ко-многим» – тип связи, где один экземпляр сущности может быть связан со множеством экземпляров другой сущности. Например, один пользователь может размещать несколько объявлений, но каждое объявление будет связано с конкретным пользователем.
3. «Многие-ко-многим» – тип связи, при котором множество экземпляров одной сущности может быть связано с множеством экземпляров другой сущности. Например, пользователи и категории товаров: один пользователь может размещать объявления в нескольких категориях, а одна категория может содержать объявления от разных пользователей. Для реализации этой связи обычно требуется создание промежуточной таблицы.

ER-диаграмма – это визуальное представление всех сущностей и их взаимосвязей, которое будет использоваться для проектирования базы данных. Диаграмма помогает детализировать структуру данных и обеспечивает понимание того, какие данные нужно хранить и как они будут взаимодействовать.

ER-модели отличаются по уровню детализации и служат для постепенного уточнения структуры данных на разных этапах проектирования. Каждый уровень модели выполняет свои задачи, начиная от общего представления системы до конкретных технических аспектов хранения данных. Выделяют три основных уровня ER-моделей:

1. Концептуальный уровень. Концептуальная модель представляет собой самую общую, высокоуровневую модель, позволяющую отобразить основную структуру и предметную область будущего проекта. Здесь выделяются основные сущности и связи, но детали, такие как атрибуты и технические характеристики, обычно отсутствуют. Концептуальный уровень ориентирован на выявление основных элементов, которые будут использоваться в приложении, и проверку того, что все термины и бизнес-объекты включены.
2. Логический уровень. Логическая ER-модель углубляет концептуальный уровень, добавляя более конкретные детали и атрибуты к сущностям. На этом этапе сущности из концептуальной модели описываются с набором характеристик, которые будут храниться в базе данных. Например, сущность "Пользователь" может включать такие атрибуты, как "Имя", "Адрес электронной почты", "Роль", а "Объявление" может содержать "Название", "Описание", "Цена".
3. Физический уровень. На физическом уровне описывается конкретная реализация базы данных. Здесь проектировщики решают, каким образом данные будут организованы и хранимы, выбирают тип СУБД (реляционная, документо-ориентированная и т.д.), указывают физические пути хранения, индексацию, типы столбцов, связи, ограничения целостности и другие специфические детали. Физическая ER-модель – это план, по которому будет разворачиваться база данных на сервере.

Таким образом, ER-модели, начиная с концептуальной, затем логической и заканчивая физической, постепенно трансформируют абстрактное представление данных в технический проект базы данных, готовой к развёртыванию в рабочем окружении.

1. СУЩНОСТИ И СВЯЗИ ПРОГРАММНОГО ПРОДУКТА

В проекте «Веб-приложение для размещения товаров и услуг» выделены следующие ключевые сущности, каждая из которых играет важную роль в структуре данных и функциональности системы:

1. Users (Пользователи): Основная сущность, включающая информацию обо всех пользователях системы. Содержит такие данные, как логин, пароль, роль, и дополнительный атрибут для идентификации принадлежности к магазину (ShopId) для продавцов.
2. Profiles (Профили): Сущность для хранения дополнительной информации о пользователях, связанная с Users. Содержит атрибуты, необходимые для персонализации и обратной связи: возраст, электронная почта и номер телефона.
3. Shops (Магазины): Сущность, представляющая магазины, в которых продавцы могут размещать свои товары. Содержит такие атрибуты, как название и описание магазина.
4. Products (Товары): Сущность, описывающая все товары, доступные в системе. Включает атрибуты: наименование, описание, цена, количество на складе, категория, фотография и связь с конкретным магазином (ShopId).
5. Carts (Корзины): Сущность, представляющая корзины, которые создаются автоматически для пользователей после их регистрации или первого входа в систему. Корзина привязана к определенному пользователю (CustomerId) и служит для временного хранения товаров, которые пользователь намеревается приобрести.
6. CartItems (Элементы корзины): Сущность, представляющая конкретные товары, добавленные в корзину. Включает атрибуты: идентификатор товара (ProductId), количество выбранных единиц и идентификатор корзины (CartId).
7. Orders (Заказы): Сущность, которая хранит данные обо всех заказах, созданных пользователями. Включает информацию о клиенте, оформившем заказ (CustomerId), а также о магазине (ShopId), что позволяет продавцам обрабатывать заказы по своим магазинам.
8. OrderItems (Элементы заказа): Сущность для хранения информации о каждом товаре, входящем в заказ. Содержит такие данные, как название магазина, описание товара, цена, количество, категория и идентификатор продукта (ProductId).

Каждая сущность включает атрибуты, обеспечивающие выполнение основных функций веб-приложения – управление товарами и услугами, взаимодействие между пользователями (покупателями и продавцами), а также создание и обработка заказов.

ER-диаграмма, представленная на рисунке 2.1, отображает взаимосвязи между перечисленными сущностями. Диаграмма помогает понять, как данные связаны между собой, и служит основой для построения базы данных, которая обеспечит эффективную реализацию веб-приложения для размещения товаров и услуг.

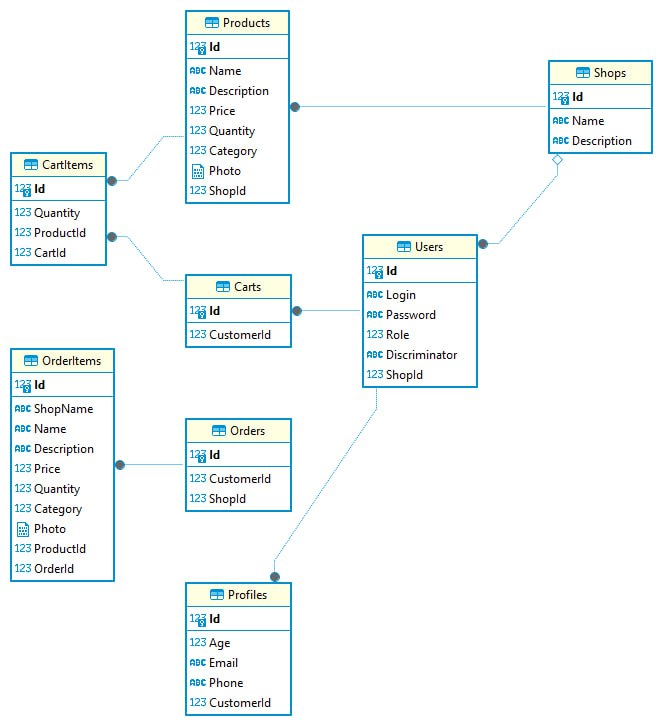


Рисунок 2.1 – ER-диаграмма базы данных

При помощи данной диаграммы можно будет отслеживать связи между объектами при разработке, а также правильно организовать структуры базы данных.

1. SQLite

Для разработки проекта «Веб-приложение для размещения товаров и услуг» была выбрана база данных SQLite в сочетании с Entity Framework.

SQLite – это легковесная встроенная реляционная база данных, которая отлично подходит для небольших и средних приложений. Она проста в использовании, не требует сложной настройки и работает на большинстве платформ, что делает ее удобным решением для разработки и тестирования. SQLite хранит данные в формате SQL, что позволяет легко управлять и анализировать структурированную информацию, такой как сведения о пользователях, товарах, заказах и других сущностях приложения.

Entity Framework (EF) – это ORM (Object-Relational Mapping) инструмент от Microsoft, который упрощает взаимодействие между приложением и базой данных. Использование EF позволяет разработчикам работать с данными на уровне объектов, не заботясь о сложностях SQL-запросов. Кроме того, Entity Framework предоставляет гибкость в работе с разными базами данных: если в будущем потребуется изменить SQLite на другую СУБД (например, SQL Server или PostgreSQL), это можно будет сделать с минимальными изменениями в коде, благодаря возможности смены базы данных в Entity Framework.

Таким образом, использование SQLite и Entity Framework обеспечивает простоту разработки, переносимость базы данных и возможность масштабирования при необходимости.