Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет   
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина «Технологии разработки программного обеспечения»

**ОТЧЕТ**

к лабораторной работе

на тему:

**«Проектирование и создание БД.»**

|  |  |
| --- | --- |
|  | Выполнил:  Слуцкий Никита Сергеевич,  студент группы 053505 |
|  | Проверил: Гриценко Никита Юрьевич, ассистент каф. Информатики |

Минск 2023

СОДЕРЖАНИЕ

[Введение 3](#_Toc146477687)

[1 Анализ существующих областей рынка 6](#_Toc146477688)

[1.1 Roomle 6](#_Toc146477689)

[1.2 RoomToDo 7](#_Toc146477690)

[1.3 Выводы и постановка задач для возможности конкурировать с вышеописанными продуктами 7](#_Toc146477691)

[2 Общее описание разрабатываемого программного продукта 9](#_Toc146477692)

[2.1 Цели разработки 9](#_Toc146477693)

[2.2 Функциональная карта программного продукта 9](#_Toc146477694)

[2.3 Пользовательские интерфейсы 11](#_Toc146477695)

[2.4 Программные интерфейсы 12](#_Toc146477696)

[2.5 Нефункциональные требования 13](#_Toc146477697)

[Список используемых источников 15](#_Toc146477698)

# Введение

Выбор и грамотное проектирование базы данных являются ключевыми аспектами разработки любого проекта, особенно в современной эпохе, где объемы данных растут экспоненциально. База данных является фундаментом системы, которая хранит и обрабатывает информацию, необходимую для бесперебойной работы приложений и эффективного управления данными.

Важность выбора и грамотного проектирования базы данных не может быть недооценена, и вот почему:

1 Эффективное управление данными: проектирование базы данных с учетом нужд проекта позволяет определить правильную структуру данных и связи между ними. Это обеспечивает эффективное хранение и доступ к данным, минимизируя дублирование и обеспечивая целостность и консистентность. Проектирование базы данных с учетом долгосрочных потребностей помогает избежать проблем с производительностью при обработке больших объемов данных.

2 Поддержка расширяемости: грамотно разработанная база данных позволяет легко расширяться в будущем. При правильном выборе и архитектуре базы данных можно добавлять новые таблицы, поля или связи с минимальными затратами на изменение существующей структуры. Это важно в ситуациях, когда проект растет или требует внесения изменений в свою функциональность.

3. Безопасность данных: проектирование базы данных с учетом безопасности позволяет обеспечить конфиденциальность и целостность данных. Адекватная аутентификация и авторизация пользователей, правильное управление правами доступа и шифрование данных помогают предотвратить несанкционированный доступ к информации и защитить конфиденциальность пользователей.

4 Интеграция и аналитика данных: грамотное проектирование базы данных упрощает интеграцию с другими системами, обеспечивая совместимость и согласованность данных. Это позволяет анализировать данные, выявлять связи и тенденции для принятия обоснованных решений. Оптимизированная база данных способствует быстрому доступу к информации и упрощает процесс агрегации и обработки данных.

5 Эффективное обслуживание и разработка: четко спроектированная база данных упрощает обслуживание и разработку приложений. Хорошо структурированные данные позволяют более легко вносить изменения в функциональность системы и обнаруживать возможные проблемы. Удобство и понятность структуры базы данных также уменьшает время, затраченное на поддержку и разработку новых функций.

В итоге, выбор и грамотное проектирование базы данных являются критическими факторами для успешной разработки проекта. Они обеспечивают эффективное управление данными, поддержку расширяемости, безопасность информации, интеграцию с другими системами и облегчают обслуживание и разработку приложений.

Целью лабораторной работы №2 является разработать структуру базы данных, создать логическую и физическую модель, построив DFD, ERD и IDEF1.X диаграммы для визуализации и представления хранимых данных. Написать DDL скрипты для будущего использования. Аргументировать выбор SQL базы данных. Подготовить отчет.

# 1 Разработка логической и физической модели данных

## 1.1 DFD диаграмма

Для начала построим DFD и рассмотрим более детально какие сущности, посредством какой информации будут взаимодействовать друг с другом. На рисунке 1 изображена диаграмма потоков данных. Рассмотрим диаграмму более детально. Ключевая сущность приложения – это пользователь: от него исходит больше всего действий. Пользователь работает с приложением под определенной ролью, от чего зависят исходящие от него запросы. Таким образом легко можно определить какие функции и как должны обрабатывать информацию, полученную от пользователя или иной сущности.

.....

## 1.2 EDE1FX диаграмма

На рисунке 2 изображена диаграмма IDEF1X, на которой выделены основные таблицы. Соединяются таблицы посредством внутренних и внешних ключей. База данных приведена к третьей нормальной форме путем устранения повторяющихся групп данных, установления связей между таблицами и устранения транзитивных зависимостей.

...

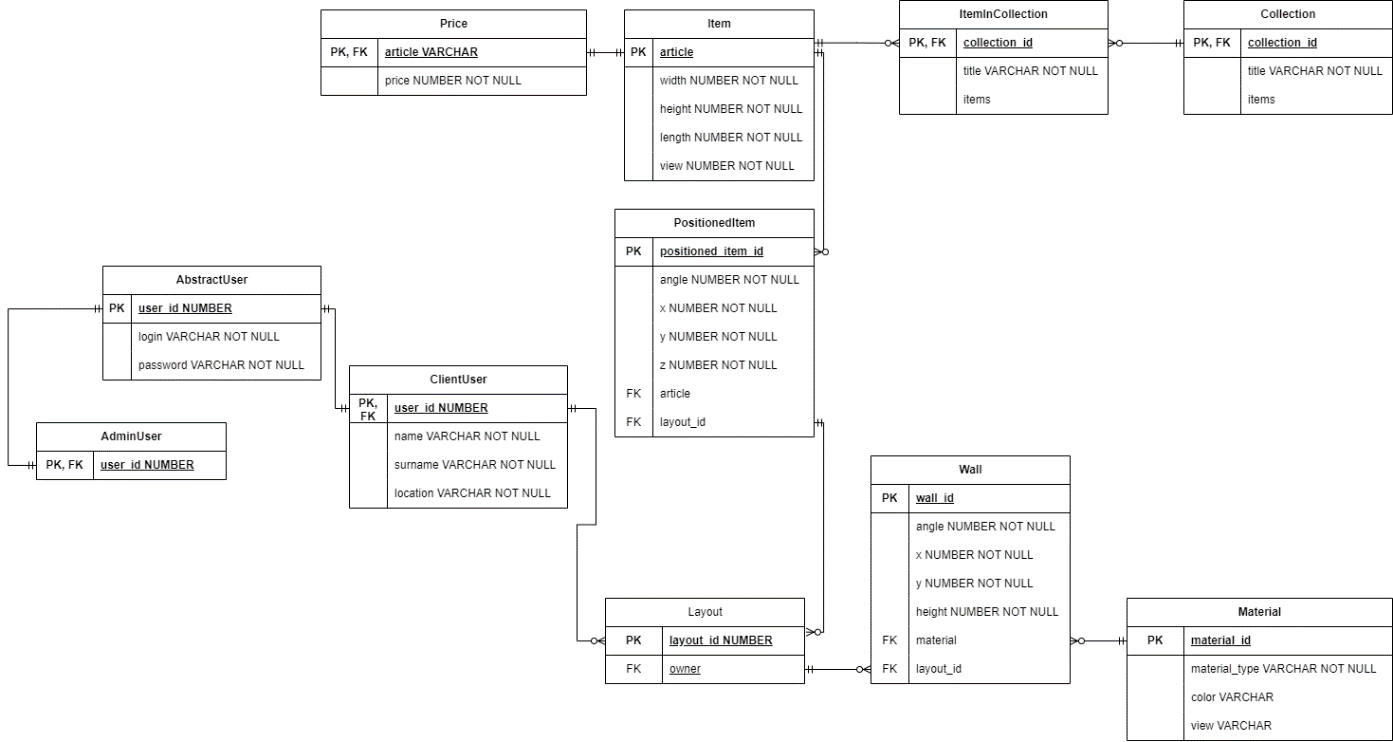


Рисунок 2 – IDE1FX диаграмма

...

## 1.3 ERD диаграмма

Теперь рассмотрим диаграмму ERD, которая изображена на рисунке 3. Диаграмма отношений сущностей позволяет рассмотреть, как построить классы, так как удобно определить: сущность включает в себя объект или коллекцию таковых.

...

# 3 Создание базы данных

## 3.1 PostgreSQL

...

# Список используемых источников

[1] 3D-планировщик [Электронный ресурс]. – Режим доступа: https://cedreo.com/faq/what-is-a-3d-floor-plan-rendering. – Дата доступа: 20.09.2023

[2] Платформа Roomle [Электронный ресурс]. – Режим доступа: https://www.roomle.com/en. – Дата доступа: 20.09.2023

[3] Платформа RoomToDo [Электронный ресурс]. – Режим доступа: https://roomtodo.com/en/. – Дата доступа: 20.09.2023

[4] Семантически корректный сайт [Электронный ресурс]. – Режим доступа: https://web.dev/learn/html/semantic-html. – Дата доступа: 21.09.2023

[4] Семантически корректный сайт [Электронный ресурс]. – Режим доступа: https://web.dev/learn/html/semantic-html. – Дата доступа: 21.09.2023

[5] Доступный интерфейс [Электронный ресурс]. – Режим доступа: https://developer.mozilla.org/en-US/docs/Learn/Accessibility/HTML/. – Дата доступа: 21.09.2023

[6] Фреймворк NextJS [Электронный ресурс]. – Режим доступа: https://nextjs.org/. – Дата доступа: 22.09.2023

[7] Язык программирования TypeScript [Электронный ресурс]. – Режим доступа: https://www.typescriptlang.org/. – Дата доступа: 22.09.2023

[8] Платформа NodeJS [Электронный ресурс]. – Режим доступа: https://nodejs.org/en. – Дата доступа: 22.09.2023

[9] Библиотека для разработки интерфейсов ReactJS [Электронный ресурс]. – Режим доступа: https://react.dev/. – Дата доступа: 22.09.2023

[10] Chrome V8 [Электронный ресурс]. – Режим доступа: https://v8.dev/. – Дата доступа: 23.09.2023

[11] СУБД MongoDB [Электронный ресурс]. – Режим доступа: https://www.mongodb.com/. – Дата доступа: 23.09.2023

[12] СУБД PostgreSQL [Электронный ресурс]. – Режим доступа: https://www.postgresql.org/. – Дата доступа: 23.09.2023

[13] IDE JetBrains WebStorm [Электронный ресурс]. – Режим доступа: https://www.jetbrains.com/webstorm/. – Дата доступа: 24.09.2023

[14] Редактор кода Microsoft Visual Studio Code [Электронный ресурс]. – Режим доступа: https://code.visualstudio.com/. – Дата доступа: 24.09.2023

[15] WebGL API [Электронный ресурс]. – Режим доступа: https://developer.mozilla.org/en-US/docs/Web/API/WebGL\_API. – Дата доступа: 24.09.2023

[16] Система контроля версий GIT [Электронный ресурс]. – Режим доступа: https://git-scm.com/. – Дата доступа: 24.09.2023