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](#_Toc146994692)

[1 Разработка логической и физической модели данных 5](#_Toc146994693)

[1.1 DFD диаграмма 5](#_Toc146994694)

[1.2 EDEF1X диаграмма 6](#_Toc146994695)

[1.3 ERD диаграмма 7](#_Toc146994696)

[2 Создание базы данных 9](#_Toc146994697)

[2.1 PostgreSQL 9](#_Toc146994698)

[2.2 Supabase 9](#_Toc146994699)

[2.3 Преимущества реляционных баз данных 10](#_Toc146994700)

[Заключение 12](#_Toc146994701)

[Список используемых источников 13](#_Toc146994702)

# 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

Диаграмма потоков данных (DFD) [1] – это графическое представление потоков данных в системе, которое показывает, как данные перемещаются между различными частями системы. DFD-диаграммы могут быть использованы для моделирования и проектирования новых систем, а также для анализа существующих систем. Они могут помочь выявить узкие места в системе и определить, какие изменения могут быть внесены для улучшения ее производительности .

Ключевая сущность приложения – это пользователь: от него исходит больше всего действий. Пользователь работает с приложением, создаёт новые лэйауты (конфигурации), вносит в них изменения, имеет возможность удалять или создавать новые, возвращаться к редактированию старых, а также получать изображения с рендерами с текущего ракурса камеры при нахождении в 3D-режиме работы конфигуратора. DFD диаграмма изображена на рисунке 1.

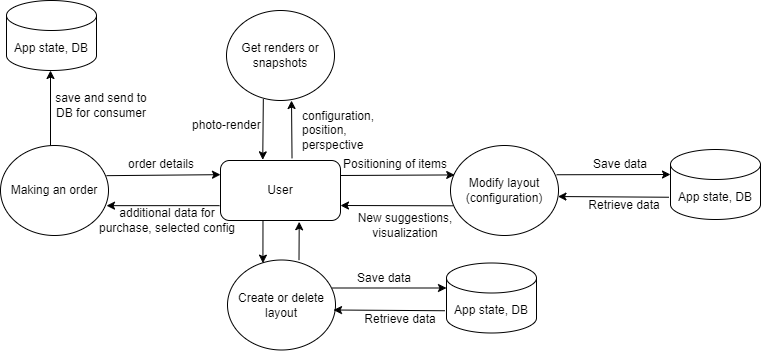


Рисунок 1 – DFD диаграмма приложения

Составление DFD-диаграммы приложения важно, потому что она помогает разработчикам и аналитикам лучше понять, как работает приложение и какие данные оно обрабатывает. Это может помочь выявить проблемы в приложении и определить, какие изменения могут быть внесены для улучшения его производительности. Кроме того, DFD-диаграммы могут быть использованы для обучения новых членов команды и для общения с заинтересованными сторонами.

## 1.2 EDEF1X диаграмма

Диаграмма IDEF1X [2] – это метод моделирования данных, который используется для разработки семантики моделей данных. Она позволяет строить семантические модели данных, которые могут служить для поддержки управления данными как ресурсом, интеграции информационных систем и построения компьютерных баз данных 1.

IDEF1X используется для формирования графических представлений информационных моделей, которые отражают структуру и семантику информации внутри среды или системы. На рисунке 2 изображена диаграмма IDEF1X, на которой выделены основные таблицы. Соединяются таблицы посредством внутренних и внешних ключей.

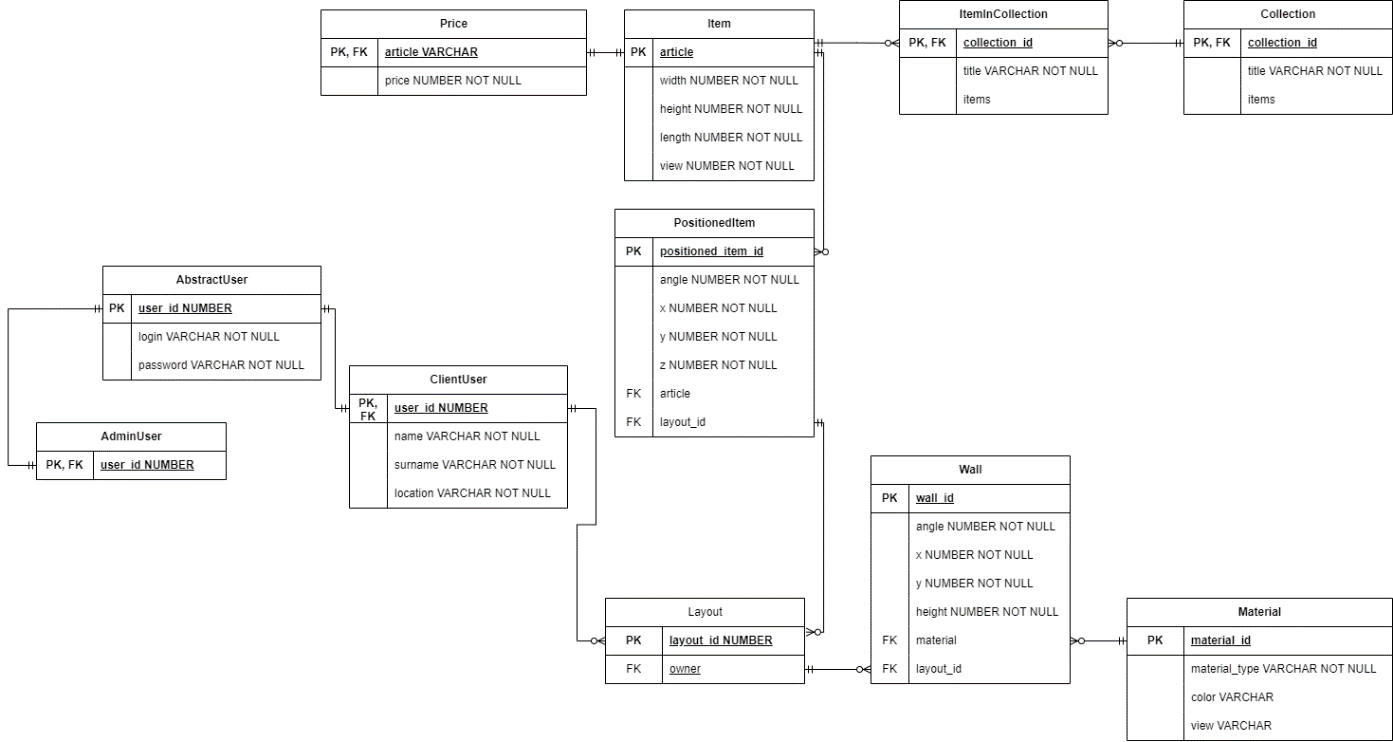


Рисунок 2 – IDEF1X диаграмма

Составление диаграммы IDEF1X приложения важно, потому что она помогает разработчикам и аналитикам лучше понять, как работает приложение и какие данные оно обрабатывает. Это может помочь выявить проблемы в приложении и определить, какие изменения могут быть внесены для улучшения его производительности. Кроме того, диаграммы IDEF1X могут быть использованы для обучения новых членов команды и для общения с заинтересованными сторонами.

Для описываемого проекта база данных спроектирована таким образом, чтобы сущности не перебивали друг друга, чтобы можно было легко добавлять новые объекты, редактировать коллекции и восстанавливать лэйауты при перезаходе в приложение.

## 1.3 ERD диаграмма

Диаграмма сущность-связь (ERD) [3] – это графическое представление сущностей и связей между ними в базе данных. Она используется для проектирования и отладки реляционных баз данных в сфере образования, исследования и разработки программного обеспечения и информационных систем для бизнеса. Существуют три уровня ERD-диаграмм: концептуальная, логическая и физическая.

Концептуальные ERD-диаграммы полагаются на стандартный набор символов, включая прямоугольники, ромбы, овалы и соединительные линии, для отображения сущностей, их атрибутов и связей. Эти диаграммы устроены по тому же принципу, что и грамматические структуры: сущности выполняют роль существительных, а связи – глаголов. Диаграмма отношений сущностей позволяет рассмотреть, как построить классы, так как удобно определить: сущность включает в себя объект или коллекцию таковых.

Она служит основой для дальнейшего проектирования базы данных. На рисунке 3 изображена концептуальная диаграмма.

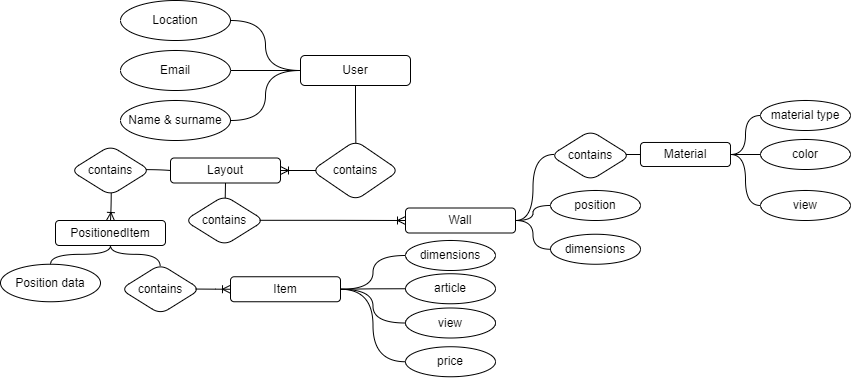


Рисунок 3 – Концептуальная ERD диаграмма

Концептуальная диаграмма является самым высоким уровнем и абстракцией ERD. Она моделирует общую структуру данных и отражает концептуальное представление информации в организации или системе. Концептуальная диаграмма позволяет выделить основные сущности, атрибуты и связи между ними, игнорируя детали реализации и ограничений баз данных.

Логическая диаграмма представляет собой более детализированное описание структуры данных, учитывающее специфику конкретной системы управления базами данных (СУБД). Она включает таблицы, столбцы, отношения между таблицами и ключи. Логическая диаграмма также может содержать ограничения целостности и другие параметры, определяющие правила редактирования и использования данных. Эта диаграмма чаще всего используется при разработке физической ERD-диаграммы. Логическая диаграмма изображена на рисунке 4.

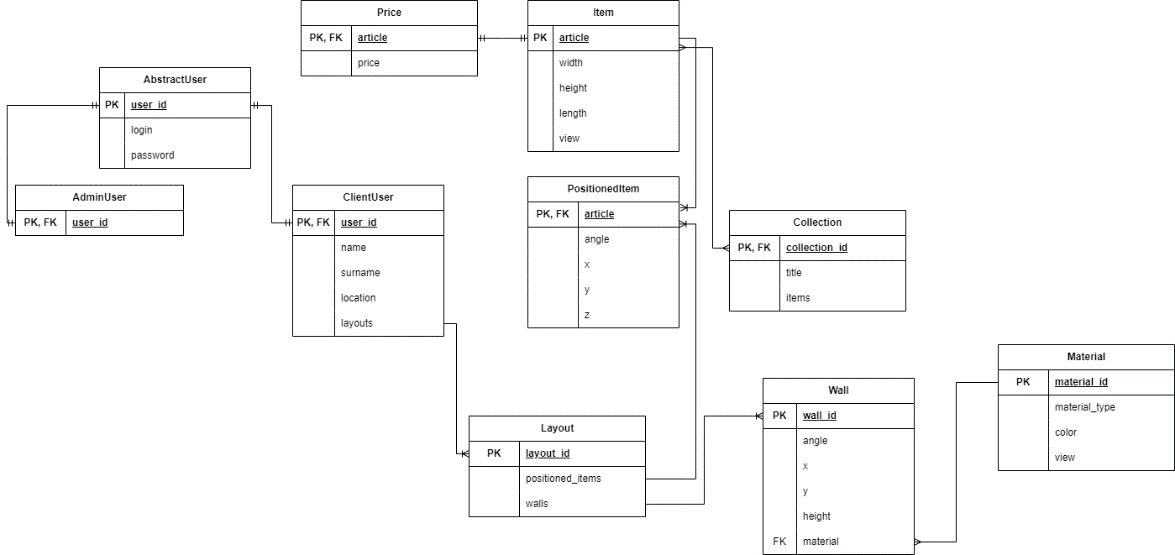


Рисунок 4 – Логическая ERD диаграмма

Физическая диаграмма представляет все необходимые детали и спецификации для реализации базы данных. Она включает точные поля, индексы, внешние ключи, типы данных и другие подробности, учитывая требования конкретной СУБД.

Физическая диаграмма является самым конкретным уровнем ERD-диаграммы и служит основой для создания таблиц и структуры базы данных. На рисунке 5 изображена физическая диаграмма ERD, на которой прописаны типы данных в соответствии с выбранной СУБД, а также связи many-to-many заменены на промежуточную таблицу.

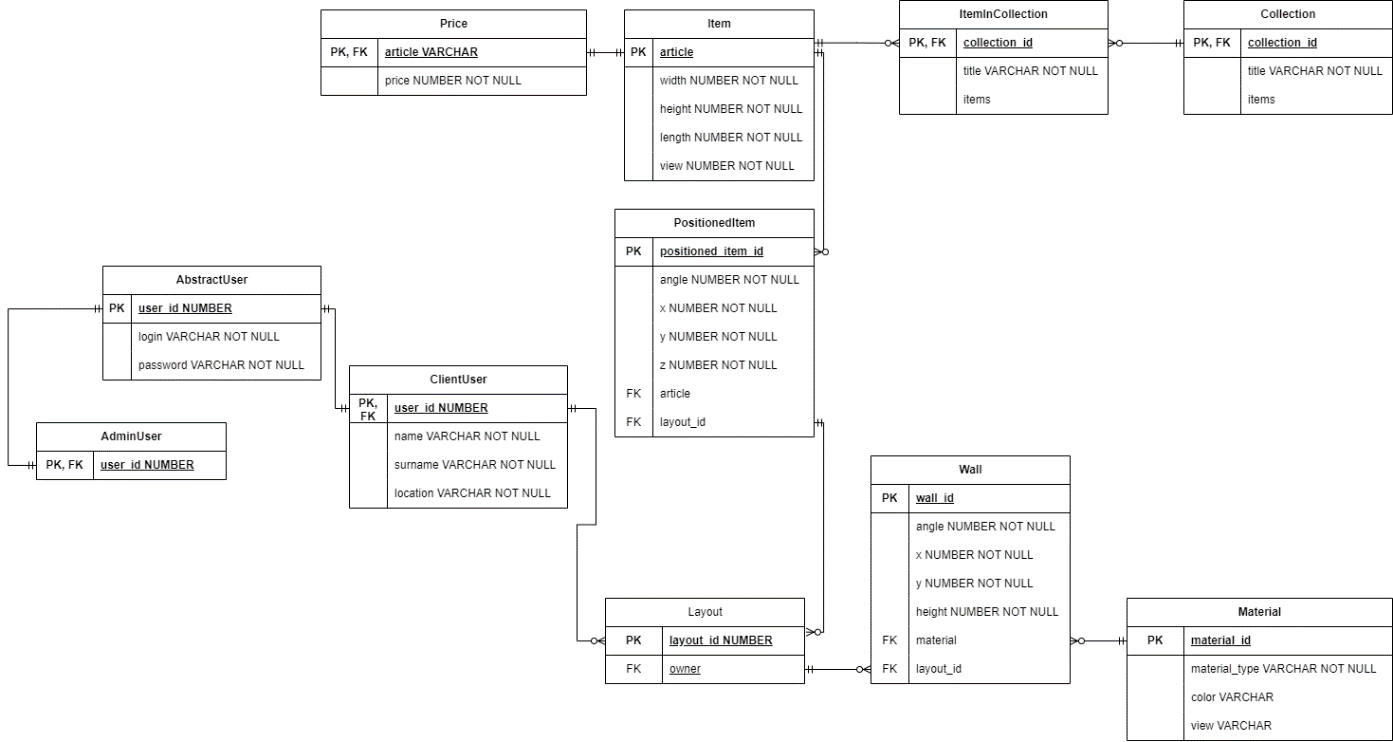


Рисунок 5 – Физическая ERD диаграмма

Составление ERD-диаграммы приложения важно, потому что она помогает разработчикам и аналитикам лучше понять, как работает приложение и какие данные оно обрабатывает. Это может помочь выявить проблемы в приложении и определить, какие изменения могут быть внесены для улучшения его производительности. Кроме того, ERD-диаграммы могут быть использованы для обучения новых членов команды и для общения с заинтересованными сторонами.

Концептуальная диаграмма фокусируется на общей структуре данных и основных понятиях, абстрагируясь от специфики реализации и ограничений СУБД. Логическая диаграмма включает более детальное описание таблиц, отношений, ключей и ограничений, учитывая специфику конкретной СУБД.

Физическая диаграмма предоставляет подробную структуру базы данных, включая поля, типы данных, индексы и другие детали, специфичные для конкретной СУБД.

Важно отметить, что ERD-диаграммы служат для визуализации и проектирования баз данных, их структур и связей. Они не являются готовыми базами данных, но предоставляют основу для их создания и разработки.

# 2 Создание базы данных

## 2.1 PostgreSQL

PostgreSQL [4] – это объектно-реляционная система управления базами данных, наиболее развитая из открытых СУБД в мире. Она имеет открытый исходный код и является альтернативой коммерческим базам данных. PostgreSQL предоставляет полный набор функций, включая расширенную поддержку SQL, сложные типы данных, сохраненные процедуры, триггеры, представления, полнотекстовый поиск, поддержку JSON и многое другое.

PostgreSQL позволяет гибко управлять базами данных. С ее помощью можно создавать, модифицировать или удалять записи, отправлять транзакцию – набор из нескольких последовательных запросов на особом языке запросов SQL.

Данная СУБД используется бэкенд-разработчиками, которым приходится взаимодействовать с базами данных при работе с серверной частью сайта, а также администраторами и разработчиками баз данных - специалистами, основная задача которых заключается в обслуживании и поддержке работоспособности базы.

## 2.2 Supabase

Supabase [5] – это платформа для создания бэкэнда, построенная на базе данных PostgreSQL. Она предоставляет полный набор функций, включая базу данных с возможностью обновления и мониторинга реального времени, аутентификацию, хранилище объектов и функции краевого вычисления. Supabase является реляционной альтернативой Firebase и предоставляет все необходимые функции бэкэнда для создания продукта.

Каждый проект в Supabase предоставляет полноценную реал-тайм базу данных PostgreSQL с уровнем доступа к PostgreSQL и управлением резервными копиями базы данных. Supabase также предоставляет несколько вариантов программного подключения к базе данных PostgreSQL: прямые подключения с использованием стандартной системы подключения PostgreSQL, пул подключений с использованием PgBouncer и программный доступ с использованием Serverless APIs 4.

Supabase используется для создания и поддержки веб- и мобильных приложений. Она может уменьшить время до маркетинга, предоставив готовый к использованию бэкэнд.

Чтобы создать базу данных в Supabase, нужно создать новый проект в Supabase Dashboard. После того, как ваш проект будет готов, создайть таблицу в базе данных Supabase можно с помощью инструментов в Dashboard. Соответственно, отпадает надобность в самостоятельном написании DDL-скриптов для создания таблиц и структуры базы данных. На рисунке 6 представлен скриншот из панели управления Supabase.

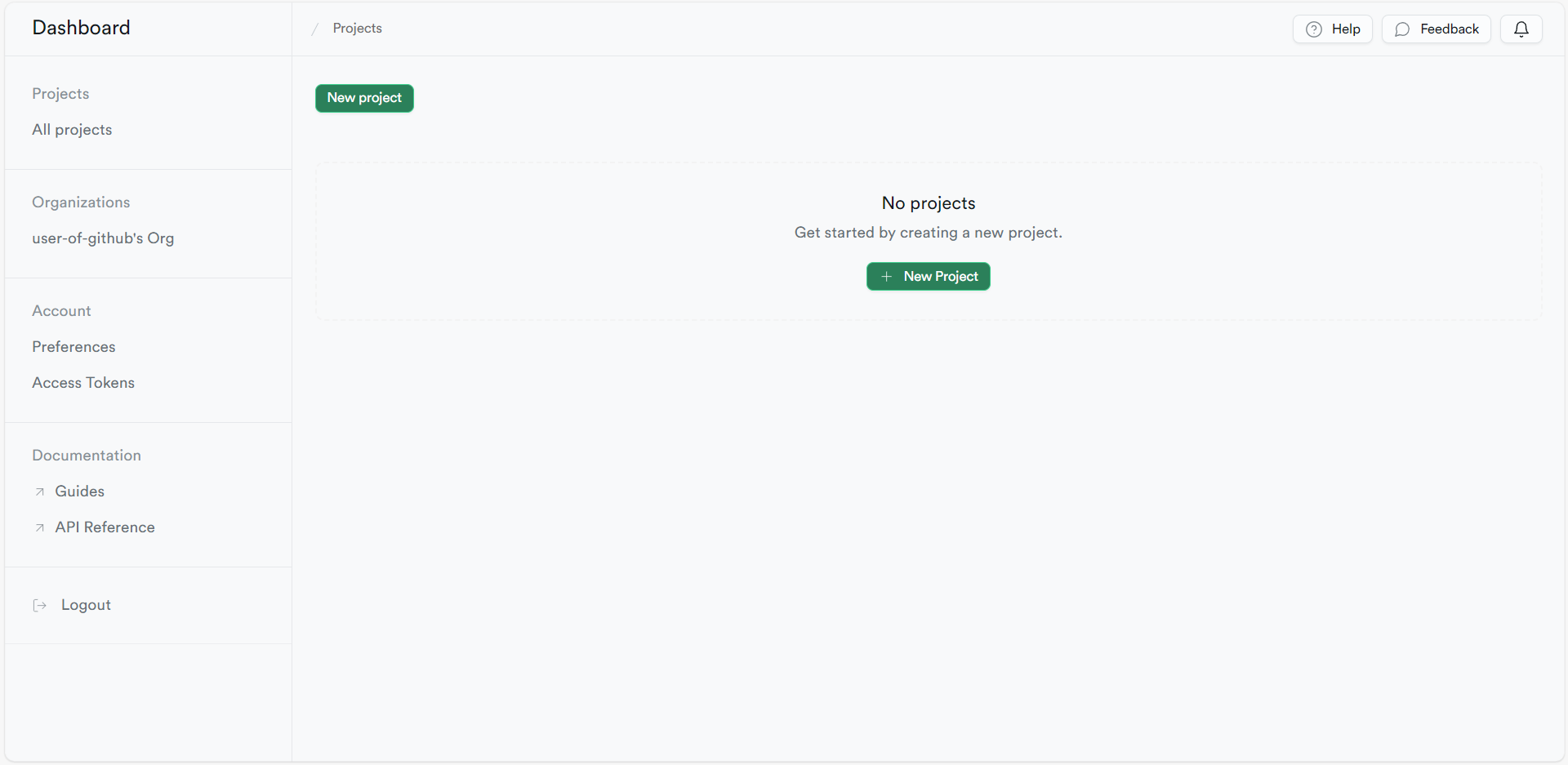


Рисунок 6 – Панель управления Supabase

Для разработки программного средства для визуальной расстановки объектов в помещении была выбрана именно эта база данных, потому что она проста в настройке и подключении, хорошо интегрируется в веб-приложения, в частности написанные в экосистеме NodeJS.

## 2.3 Преимущества реляционных баз данных

Реляционная модель данных – это логическая модель данных, которая является одним из подходов к организации данных в программном обеспечении баз данных. Она использует таблицы для хранения и организации данных, где каждая таблица состоит из строк и столбцов. Каждая строка в таблице представляет собой отдельную запись, а каждый столбец представляет собой отдельный атрибут или характеристику записи.

Преимущества реляционной модели:

1 Простота использования: реляционная схема является простой и интуитивно понятной моделью, которая легко понимается и используется большинством разработчиков баз данных.

2 Гибкость: реляционная модель позволяет легко добавлять, изменять и удалять данные в таблицах, что делает ее гибкой и удобной для использования.

3 Надежность: обеспечивает высокую степень надежности и целостности данных благодаря использованию ограничений целостности и транзакций.

4 Эффективность: такая модель обеспечивает быстрый доступ к данным благодаря использованию индексов и оптимизации запросов.

# Заключение

В результате выполнения лабораторной работы №2 была разработана структура базы данных, создана логическая и физическая модель, построены DFD, ERD и IDEF1.X диаграммы для визуализации и представления хранимых данных, а также проведён анализ использованных диаграмм.

Ниже представлены некоторые основные различия между ERD, DFD и IDEF1X диаграммами. Целью ERD является моделирование структуры данных и связей между сущностями (entities) в системе. DFD визуализирует потоки данных между различными процессами и сущностями в системе, показывая, как данные перемещаются. IDEF1X: Целью IDEF1X является моделирование структуры данных, основываясь на концепции сущностей, атрибутов и связей, чтобы создать нормализованную БД.

ERD может иметь разные уровни детализации, начиная от концептуальной диаграммы, описывающей общую структуру данных, до физической диаграммы, включающей конкретные таблицы и связи.

DFD также может иметь разные уровни детализации, начиная с высокоуровневого уровня контекстной диаграммы, описывающей общий взгляд на систему, до более детализированных диаграмм, показывающих отдельные процессы и взаимодействия.

IDEF1X обычно представляет детализированную структуру данных, с фокусом на сущностях, атрибутах, отношениях и ключах.

Все эти типы диаграмм имеют свои преимущества и применяются в разных ситуациях, в зависимости от того, на что вы хотите сосредоточиться при моделировании системы. Они важны для понимания, проектирования и разработки эффективных информационных систем.

Создана база данных. В генерации DDL скриптов отсутствует необходимость, потому что выбранная база данных Supabase позволяет более автоматизировано создавать сущности в базе. Аргументирован выбор реляционной базы данных и подготовлен отчет.

Цели лабораторной работы можно считать достигнутыми.

# Список используемых источников

[1] DFD-диаграмма [Электронный ресурс]. – Режим доступа: https://habr.com/ru/articles/668684/. – Дата доступа: 27.09.2023

[2] IDEF1X-диаграмма [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/IDEF1X. – Дата доступа: 28.09.2023

[3] ERD-диаграмма [Электронный ресурс]. – Режим доступа: https://www.lucidchart.com/pages/ru/erd-диаграмма. – Дата доступа: 29.09.2023

[4] Реляционная база данных PostgreSQL [Электронный ресурс]. – Режим доступа: https://www.postgresql.org/. – Дата доступа: 30.09.2023

[5] База данных Supabase [Электронный ресурс]. – Режим доступа: https://supabase.com/. – Дата доступа: 30.09.2023