Санкт-Петербургское государственное бюджетное

профессиональное образовательное учреждение

«Ижорский колледж»

КУРСОВАЯ РАБОТА

ТЕМА: «Платформа для онлайн-опросов»

по ПМ.11 МДК.11.01 Технология разработки и защиты баз данных

Выполнил

обучающийся группы 231с

Бабич Игорь Владимирович

Проверил

преподаватель спец. дисциплин

специальности 09.02.07

Информационные системы и программирование

Венедиктов Д.В.

Санкт-Петербург, 2025

ЗАДАНИЕ ДЛЯ ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

1. Изучить теоретические основы технологии разработки баз данных, включая этапы проектирования, нормализации, реализации и тестирования.

2. Рассмотреть методы и средства защиты баз данных от несанкционированного доступа, атак и вредоносных программ.

3. Провести анализ существующих систем управления базами данных (СУБД) с точки зрения их функциональности, безопасности и удобства использования.

4. Разработать проект базы данных для конкретной предметной области, учитывая требования к безопасности и конфиденциальности данных.

5. Реализовать разработанный проект в выбранной СУБД.

6. Протестировать разработанную базу данных на предмет соответствия требованиям безопасности и надёжности.

7. Оценить эффективность разработанной системы защиты базы данных и предложить рекомендации по её улучшению.

8. Оформить результаты работы в виде курсовой работы, включающей введение, основную часть, заключение, список использованных источников и приложения.

В основной части курсовой работы необходимо подробно рассмотреть следующие вопросы:

* основные этапы разработки базы данных;
* методы обеспечения целостности и безопасности данных;
* механизмы аутентификации и авторизации пользователей;
* шифрование данных и защита от вредоносного ПО;
* особенности реализации проекта базы данных в выбранной СУБД;
* результаты тестирования и оценка эффективности системы защиты.

Для выполнения задания рекомендуется использовать современные научные источники, а также практический опыт разработки и защиты баз данных.

ПЛАН-ГРАФИК ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

|  |  |  |
| --- | --- | --- |
| Дата | Содержание работ | Отметка о выполнении |
| 10.09 | Выбор темы курсовой работы |  |
| 17.09 | Знакомство с методическими рекомендациями по выполнению курсовой работы |  |
| 24.09 | Знакомство с планом-графиком выполнения курсовой работы |  |
| 08.10 | Работа с нормативно-правовыми документами, учебной литературой |  |
| 22.10 | Анализ предметной области. Постановка задачи |  |
| 29.10 | Составление введения к курсовой работе |  |
| 05.11 | Описание предметной области и функции решаемых задач |  |
| 12.11 | Выбор средств для выполнения курсовой работы. Выбор среды разработки |  |
| 19.11 | Концептуально-логическое проектирование. Составление ER-диаграммы |  |
| 10.12 | Создание и заполнение базы данных |  |
| 24.12 | Представления в базе данных |  |
| 21.01 | Процедуры в базе данных |  |
| 04.02 | Создание ролей в базе данных |  |
| 18.02 | Аутентификация пользователей в базе данных |  |
| 25.02 | Импорт и экспорт базы данных |  |
| 04.03 | Разработка стратегии резервного копирования базы данных |  |
| 11.03 | Тестирование базы данных |  |
| 18.03 | Оптимизация базы данных |  |
| 25.03 | Составление заключения к курсовой работе |  |
| 01.04 | Составление списка источников информации, используемых при выполнении курсовой работы |  |
| 08.04 | Подготовка курсовой работы к защите |  |

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 6

ГЛАВА 1. ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ БАЗЫ ДАННЫХ 7

1.1. Анализ предметной области 7

1.2. Анализ технологий для разработки базы данных 8

1.3. Постановка задачи курсовой работы 10

1.3.1. Цели курсовой работы 10

1.3.2. Задачи курсовой работы 11

1.4.1. Функциональные требования к системе 11

1.4.2. Нефункциональные требования к системе 11

1.5. Выбор программных средств для разработки базы данных 11

1.6. Выбор среды для разработки базы данных 12

ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ 13

2.1. Концептуально-логическое моделирование 13

2.1.1 Концептуальное моделирование 13

2.1.2. Логическое моделирование 16

2.2. Описание информационных объектов базы данных 20

2.3.1. Создание базы данных 22

2.3.2. Заполнение базы данных 23

2.4. Представления в базе данных 27

2.5. Процедуры в базе данных 31

2.6. Создание ролей в базе данных 33

2.7. Аутентификация пользователей в базе данных 33

2.8. Импорт и экспорт базы данных 38

2.9. Разработка стратегии резервного копирования базы данных 41

2.10.1. Тестирование базы данных 41

2.10.2. Оптимизация базы данных 44

ЗАКЛЮЧЕНИЕ 45

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 46

ВВЕДЕНИЕ

В современном цифровом мире данные представляют собой один из ключевых ресурсов, влияющих на стратегические решения и операционные процессы организаций. Базы данных стали основным инструментом для хранения, обработки и управления информацией, обеспечивая эффективное взаимодействие с данными. Разработка баз данных – это сложный процесс, требующий комплексного подхода, включающего анализ требований, проектирование, реализацию и поддержку.

С учетом растущего объема информации и усложнения бизнес-процессов, значимость качественного проектирования баз данных трудно переоценить. Правильная структура базы данных обеспечивает не только ее функциональность и производительность, но и легкость в управлении данными, что в свою очередь влияет на оперативность принятия решений.

Однако с увеличением объема данных и многообразием технологий, связанных с их хранением, возникает необходимость уделять особое внимание защите данных. Утечки информации и кибератаки могут привести к значительным финансовым потерям и подрыву доверия со стороны клиентов. Поэтому технологии защиты баз данных становятся критически важными для бизнеса всех размеров.

Эта работа сосредоточится на ключевых аспектах разработки и защиты баз данных, включая методы проектирования и элементы безопасности, которые должны быть внедрены для обеспечения надежности и защиты данных в любой организации. Изучение этих тем поможет определить лучшие практики, которые должны применяться для достижения устойчивости и эффективности работы с информацией в условиях динамично меняющейся технологической среды.

ГЛАВА 1. ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ БАЗЫ ДАННЫХ

1.1. Анализ предметной области

Технология разработки и защиты баз данных охватывает широкий спектр процессов, методов и инструментов, необходимых для создания, управления и обеспечения безопасности баз данных. Это область, которая активно развивается в ответ на растущие требования к хранению и защите информации, что делает её актуальной для различных отраслей, таких как финансы, здравоохранение, образование и многие другие.

1.2. Анализ технологий для разработки базы данных

Разработка баз данных включает в себя использование различных технологий и инструментов, которые помогают осуществлять проектирование, реализацию, управление и оптимизацию баз данных. Рассмотрим основные технологии и подходы, которые широко применяются в этой сфере.

**Системы управления базами данных (СУБД)**

СУБД являются основными инструментами для создания и управления базами данных. Они могут быть разделены на несколько категорий:

* реляционные СУБД (RDBMS): MySQL, PostgreSQL, Oracle, Microsoft SQL Server;
* нереляционные СУБД (NoSQL): MongoDB, Cassandra, Redis;
* гибридные СУБД: Microsoft Azure Cosmos DB.

**Языки программирования**

* SQL (Structured Query Language): основной язык для работы с реляционными базами данных. Позволяет создавать, изменять, запрашивать и управлять данными;
* Python и JavaScript: часто используются для взаимодействия с базами данных в веб-приложениях. Библиотеки, такие как SQLAlchemy или Sequelize, упрощают работу с базами данных;
* Java и C#: используются в крупных корпоративных приложениях, особенно с реляционными базами данных.

**Инструменты для проектирования баз данных**

* ER-моделирование: Инструменты, такие как Lucidchart, Draw.io или ER/Studio, позволяют создавать диаграммы сущностей и взаимосвязей (ER-диаграммы), которые помогают визуализировать структуру базы данных;
* системы управления версиями: Git и другие системы позволяют контролировать изменения в схемах баз данных и процессах миграции.

**Инструменты для обеспечения безопасности баз данных**

* шифрование: использование технологий шифрования данных (например, AES) для защиты информации как «в покое», так и «в движении»;
* аутентификация и авторизация: применение OAuth, OpenID Connect и других протоколов для управления доступом к базе данных;
* мониторинг и аудит: инструменты, такие как Splunk, могут использоваться для отслеживания запросов и анализа событий безопасности.

**Методы оптимизации производительности**

* индексация: создание индексов для ускорения выполнения запросов, особенно в реляционных базах данных.
* кэширование: использование кэшей, таких как Redis или Memcached, для ускорения доступа к часто запрашиваемым данным.
* балансировка нагрузки: распределение запросов между несколькими серверами баз данных для повышения производительности и отказоустойчивости.

Выбор технологий для разработки базы данных зависит от специфики проекта, объема данных, ожидаемых нагрузок и требований к безопасности. Комплексный подход к выбору инструментов и методов способствует созданию эффективных и надежных систем управления данными, которые могут адаптироваться к изменяющимся условиям и требованиям.

1.3. Постановка задачи курсовой работы

С ростом объемов данных, а также увеличением угроз безопасности информации, эффективная разработка и защита баз данных становятся крайне важными задачами для организаций. Безопасность данных имеет критическое значение для решения вопросов конфиденциальности, целостности и доступности информации.

Успешное выполнение работы будет способствовать более глубокому пониманию актуальных проблем и новых подходов в области безопасного управления данными.

1.3.1. Цели курсовой работы

Основная цель заключается в создании эффективной и безопасной базы данных, способной обеспечить надежный доступ к информации и защиту от потери или несанкционированного доступа.

1.3.2. Задачи курсовой работы

* проектирование архитектуры баз данных;
* разработка схемы данных и моделей;
* обеспечение безопасности данных через шифрование, аутентификацию и авторизацию;
* реализация механизмов резервного копирования и восстановления;
* мониторинг и аудит доступа к данным.

1.4.1. Функциональные требования к системе

* возможность создания, обновления и удаления данных;
* поддержка сложных запросов и отчетов;
* обеспечение многопользовательского доступа.

1.4.2. Нефункциональные требования к системе

* высокая производительность и надежность;
* защита от несанкционированного доступа;
* удобный интерфейс для пользователей.

1.5. Выбор программных средств для разработки базы данных

Выбор программных средств для разработки баз данных является ключевым этапом, который влияет на функциональность, производительность и безопасность системы.

При выполнении данной курсовой работы я планирую использовать следующие программные средства:

* система управления базами данных – MySQL;
* языки программирования – JavaScript, SQL, Java;
* средства проектирования базы данных – Lucidchart, Liquibase;
* инструменты для обеспечения безопасности данных – AWS KMS, Splunk Enterprise Security;
* средства тестирования и оптимизации приложений – Selenium JMete, EXPLAIN.

Выбор программных средств для разработки базы данных зависит от специфических требований проекта, таких как объем данных, тип доступа, требования к безопасности и производительности. Важно учитывать будущую масштабируемость и поддерживаемость решений, чтобы создать надежную и эффективную систему.

1.6. Выбор среды для разработки базы данных

Выбор подходящей среды для разработки базы данных является критически важным этапом проекта, который может оказать значительное влияние на эффективность работы команды и качество конечного продукта.

При выполнении данной курсовой работы я планирую использовать следующие среды и инструменты:

* IDE и редактор кода – MySQL;
* система контейнеризации – Docker;
* система управления версиями – git;
* фреймворк и библиотека – Hibernate;

Выбор среды для разработки базы данных требует комплексного подхода с учетом специфики проекта, требований к производительности, безопасности и интеграции. Правильный выбор инструментария и технологии существенно влияет на успех проекта, поэтому следует тщательно проанализировать свои потребности и доступные решения.

ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

2.1. Концептуально-логическое моделирование

Концептуально-логическое моделирование – это важный этап в проектировании базы данных, который позволяет структурировать данные и определить их взаимосвязи. Этот процесс разделяется на два ключевых этапа: концептуальное моделирование и логическое моделирование.

2.1.1 Концептуальное моделирование

Целью концептуального моделирования является определение основных сущностей и их связей в предметной области, не вдаваясь в детали реализации.

**Идентификация сущностей**

* Users
* Surveys
* Questions
* Response
* Lamentation

**Определение атрибутов**

Users: ID, Password, Email, Role

Surveys: ID, Title, Desctiption, Status, StartDate, EndDate, IsAnonymus, UserID.  
 Questions: ID, Text, Type, IsRequired, Order, SurveyID.

Response: ID, SubmissionDate, UserId, SurveyID, SessionID, Duration.

Lamentation: ID, Description, CreatedAt, UpdatedAt, SurveyID, UserID, Status.

**Установление связей**

* **Users → Surveys**
  + Один пользователь может создать **много опросов**.
  + Связь: Users.ID → Surveys.UserID (1:N).
* **Surveys → Questions**
  + Один опрос содержит **много вопросов**.
  + Связь: Surveys.ID → Questions.SurveyID (1:N).
* **Questions → Response**
  + На один вопрос может быть **много ответов**.
  + Связь: Questions.ID → Response.QuestionsID(1:N).
* **Users → Response**
  + Пользователь может оставить **много ответов** (если авторизован).
  + Связь: Users.ID → Response.UserId (1:N).
* **Lamentation→ Surveys**
  + Одна жалоба может относиться к одному опросу
  + Связь: Survey.ID → Lamentation.SurveyID (1:N).
* **Lamentation→ User**
  + Один пользователь может отправить много жалоб
  + Связь: User.ID → Lamentation.UserID (1:N).

**Создание ER-диаграммы**

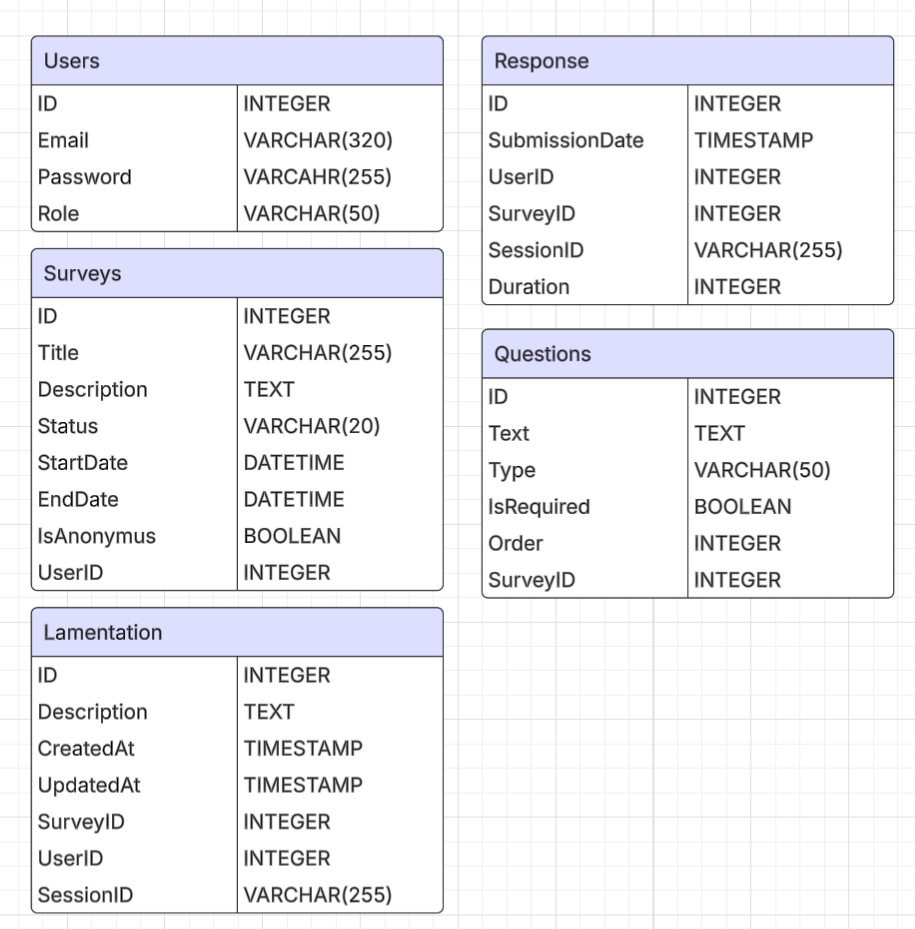


Рисунок 1 – ER-диаграмма разрабатываемой базы данных

2.1.2. Логическое моделирование

Целью логического моделирования является перевод концептуальной модели в логическую, с уточнением структуры базы данных и обязательств для хранения данных.

**Определение первичных ключей**

* Users: ID
* Surveys: ID
* Questions: ID
* Response: ID
* Lamentation: ID

**Формализация связей**

* Для связи между пользователем и опросом нужен: UserID.
* Для связи между опросом и вопросом нужен: SurveyID.
* Для связи между ответом и опросом нужен: SurveyID, QestsID.
* Для связи между ответом и пользователем нужен: UserID.
* Для связи между ответом и пользователем, если опрос анонимен нужен: SessionID.
* Для связи между жалобой и пользователем используется: UserID.
* Для связи между жалобой и опросом используется: SurveyID.

**Определение типов данных**

Users:

ID: INTEGER

Email: VARCHAR(320)

Password: VARCHAR(255)

Role: VARCHAR(50)

Surveys:

ID: INTEGER

Title: VARCHAR(255)

Description: TEXT

Status: VARCHAR(20)

StartDate: DATETIME

EndDate: DATETIME

IsAnonymus: BOOLEAN

UserID: INTEGER

Response:  
 ID: INETGER  
 SubmissionDate: TIMESTAMP

UserID: INTEGER

SureveyID: INTEGER

QuestsID: INTEGER

SessionID: VARCHAR(255)

Duration: INTEGER

Questions:  
 ID: INTEGER

Text: TEXT

Type: VARCAHR(50)

IsRequired: BOOLEAN

Order: INTEGER

SurveyID: INTEGER

Lamentation:

ID: INTEGER

Description: TEXT

Status: VARCHAR(50)  
CreatedAt: TIMESTAMP

UpdatedAt: TIMESTAMP

SurveyID: INTEGER

UserID: INTEGER  
SessionID:VARCHAR(255)

**Создание логической схемы базы данных**

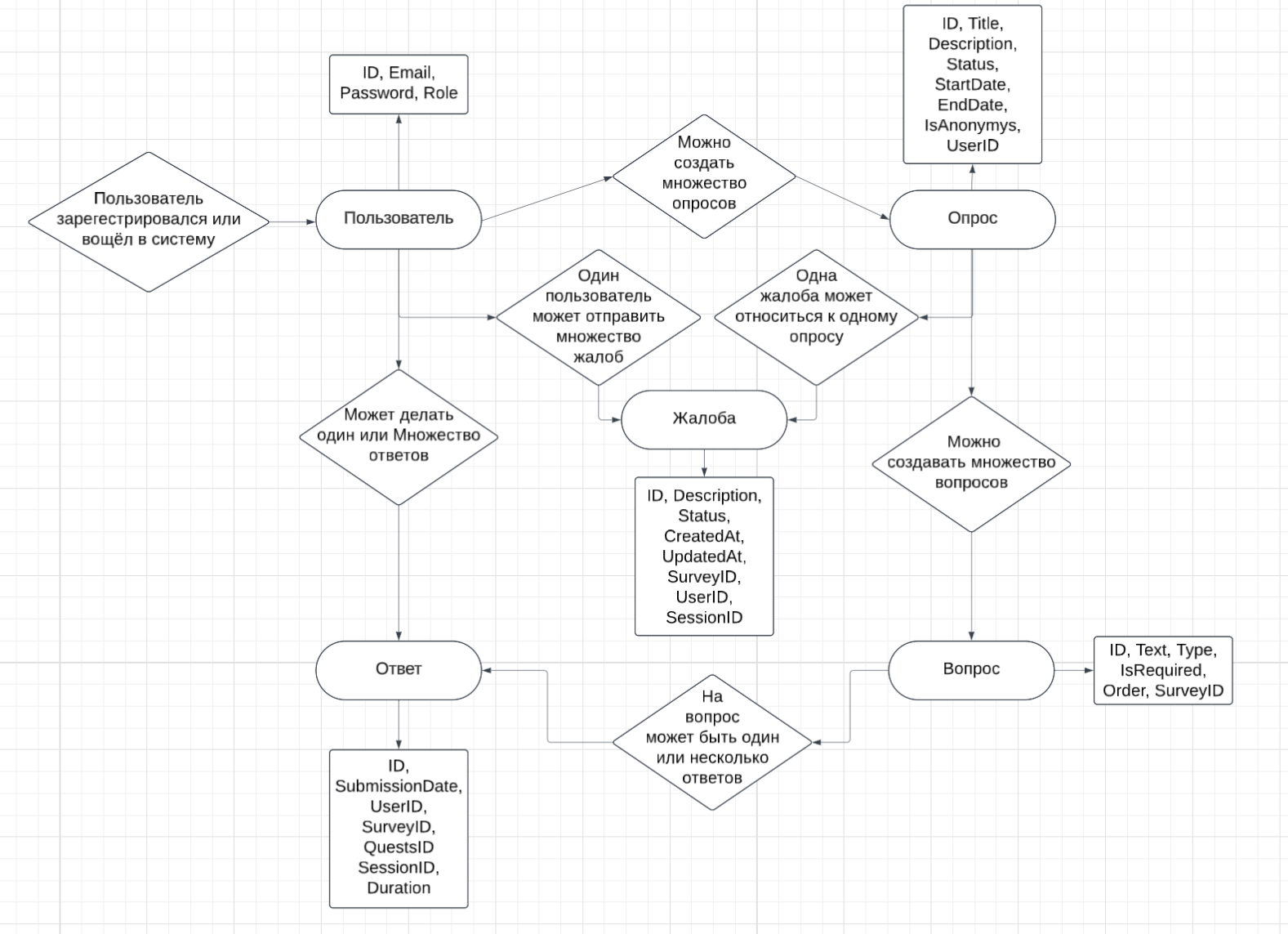


Рисунок 2 – Логическая схема базы данных

Концептуально-логическое моделирование является важным этапом разработки базы данных, который позволяет углубиться в детализацию структуры данных и обеспечить целостность и согласованность информации. Правильное моделирование позволяет избежать проблем на следующих этапах разработки и эксплуатации базы данных.

2.2. Описание информационных объектов базы данных

Информационные объекты базы данных представляют собой основные строительные блоки, на которых базируется структура любой базы данных. Они включают в себя различные сущности и их атрибуты, которые позволяют организовать, хранить и управлять данными.

Таблица 1 – информационные объекты разрабатываемой базы данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Объект | Атрибуты | Связи | Ограничения | Тип данных |
| Пользователь | ID, Email, Password. | Один пользователь может создавать много опросов. Один пользователь может пройти определённый опрос только 1 раз. | Почта должна быть уникальной, пароль должен быть обязателен | INTEGER, VARCHAR |
| Опросы | ID, Title, Description, Status, StartDate, EndDate, IsAnonymus, UserID. | Опрос, может иметь множество вопросов. | обязательны титл и вопрос | INTEGER, VARCHAR, DATETIME, BOOLEAN |
| Вопросы | ID, Text, Type, IsRequired, Order, SurveyID. | Вопрос может иметь множество вариантов ответов, может иметь разные типы ответов. | Обязательно поле текста.  Вопрос должен иметь хотя бы один ответ | INTEGER, VARCHAR, BOOLEAN, TEXT |
| Ответ | ID, SubmissionDate, UserID, SurveyID, QuestsID, SessionID, Duration | Ответ делает пользователь, на один вопрос.  Необходим один ответ, если вопрос обязательный | Если опрос анонимный, то используется не UserID, а SessionID | INTEGER, VARCHAR, TIMESTAMP, |
| Жалобы | ID, Description, Status, CreatedAt, UpdatedAt, SurveyID, UserID. | Жалоба относится к пользователю и опросу. На жалобу могут откликнуться только пользователи с ролью “admin”. | Поле Description обязательно к заполнению | INTEGER, VARCHAR, TIMESTAMP |

Описание информационных объектов базы данных позволяет структурировать и организовать информацию, обеспечивая эффективное хранение и доступ к данным. Правильное определение сущностей, их атрибутов и связей является ключом к успешному проектированию базы данных и ее функциональной полноте.

2.3.1. Создание базы данных

Для того что бы создать базу данных можно использовать консольные SQL команды, которые работают всместе с интерфейсом phpMyAdmin, создаем базу данных с обозначительным именем курсовой работы

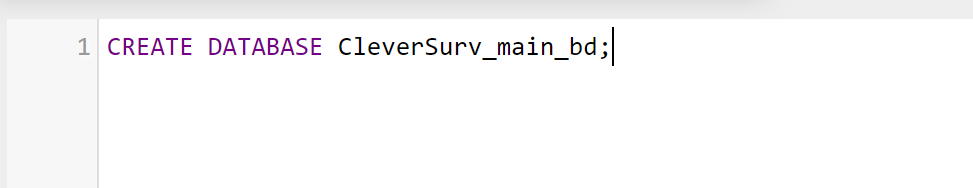


Рисунок 3 – созданная база данных

2.3.2. Заполнение базы данных

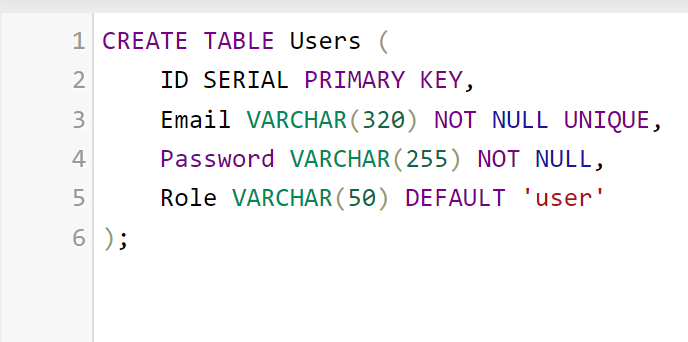
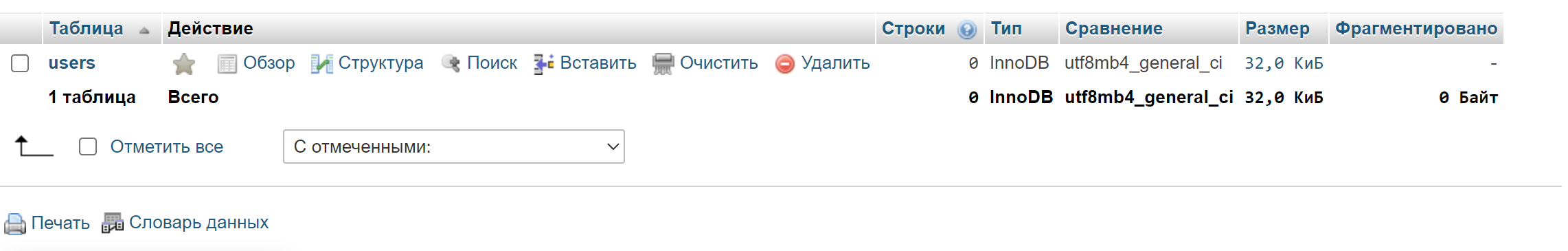


Рисунок 4 – заполнение базы данных

После успешного ответа от SQL-запроса можно увидеть таблицу в нашей базе данных.

Рисунок 5 – добавленные объекты в базу данных

По такой же аналогии были созданы еще три таблицы: «Survays», «Questions», «Response», « Lamentation» - которые представляют из себя такие же объекты.

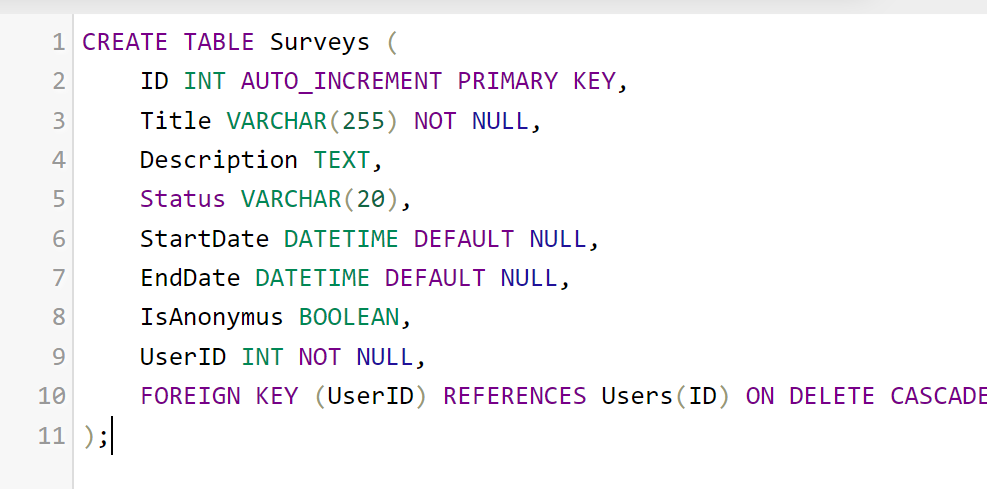


Рисунок 6 – созданная таблицы «Surveys»

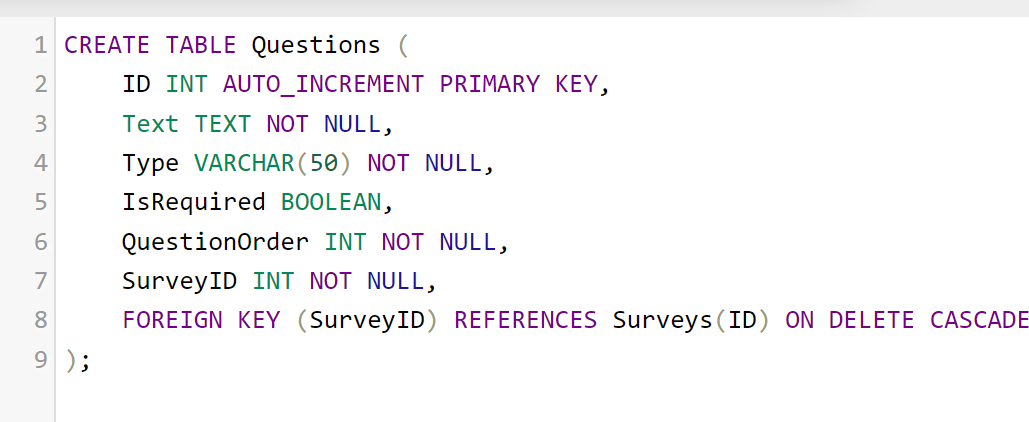


Рисунок 7 – созданная таблицы «Questions»

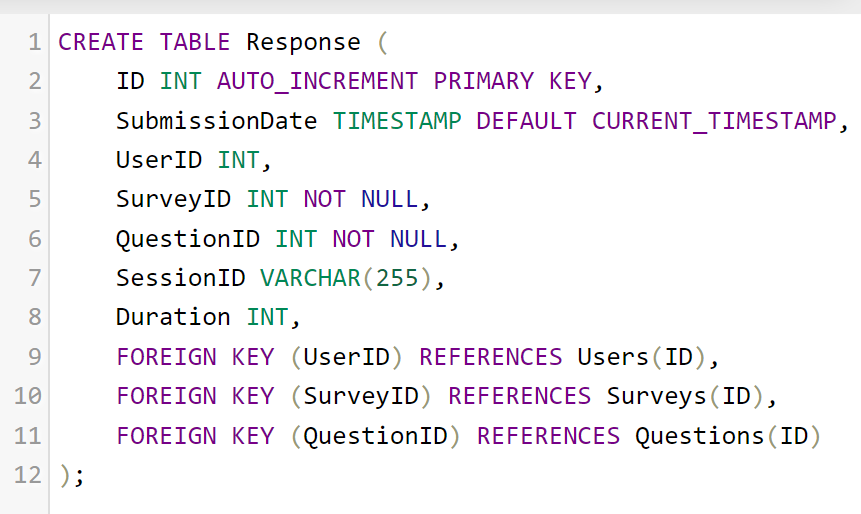


Рисунок 8 – созданная таблицы «Response»

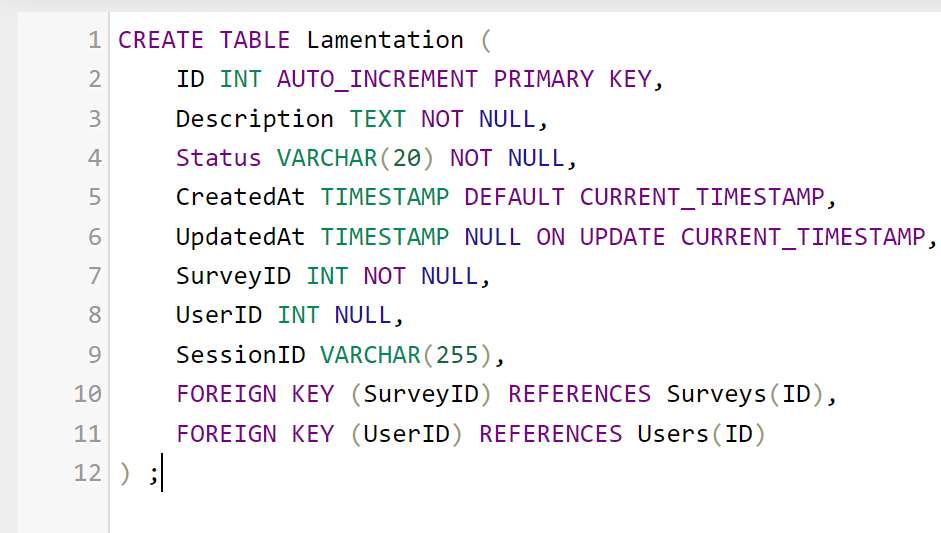


Рисунок 9 – созданная таблицы «Lamentation»

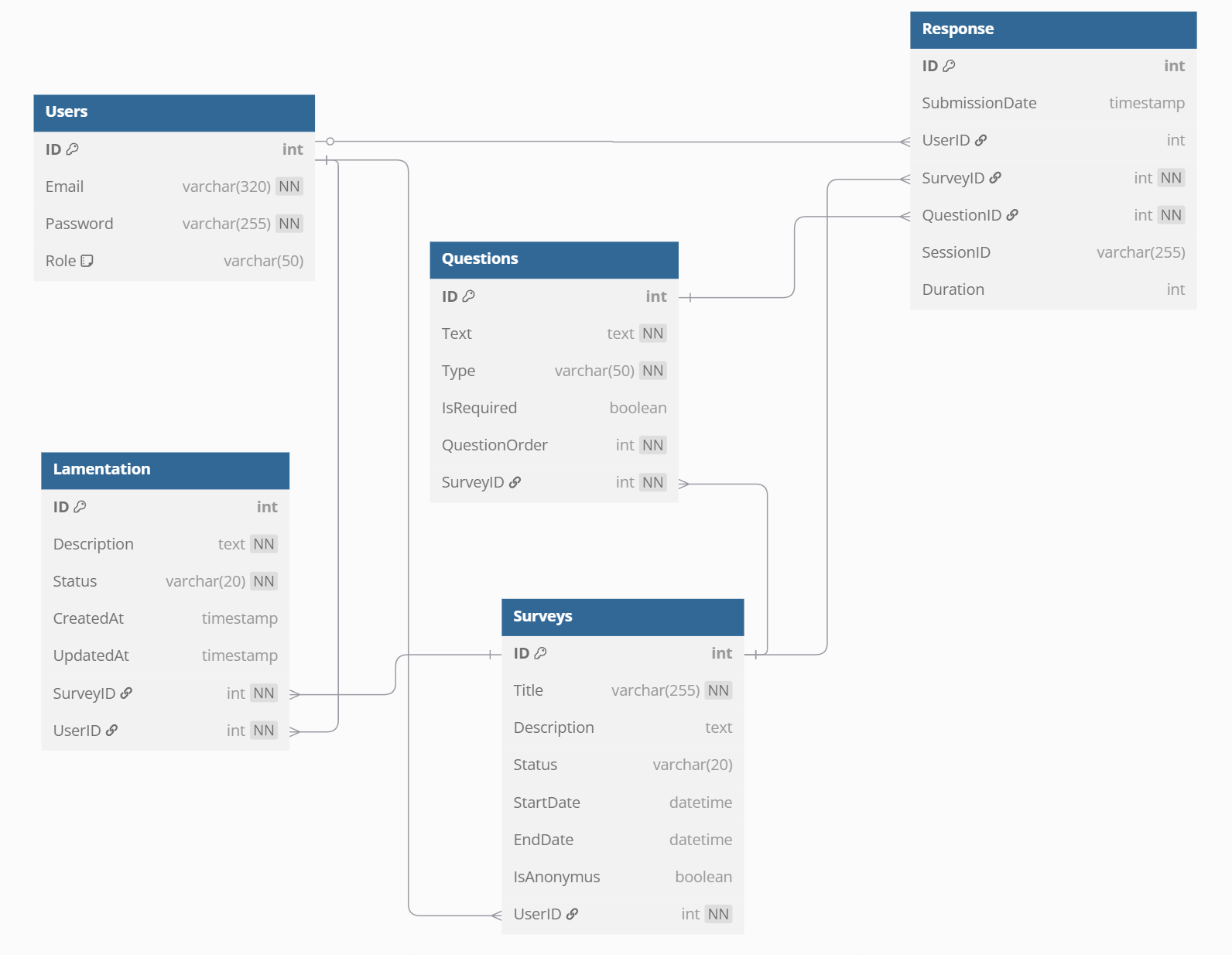


Рисунок 10 – установление связей между объектами в базе данных

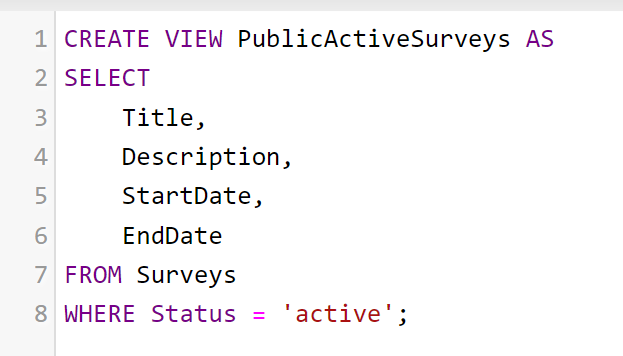
2.4. Представления в базе данных

Представление в базе данных – это виртуальная таблица, которая формируется на основе результата выполнения SQL-запроса. Она не содержит данных, а лишь определяет, как данные будут отображаться пользователям.

Функции представлений:

* упрощение сложных запросов – позволяют объединять данные из нескольких таблиц и показывать только нужные поля, что делает работу с данными более удобной;
* безопасность – можно ограничить доступ пользователей к определённым столбцам или строкам баз данных, предоставляя доступ только к представлениям;
* логическая независимость – изменения в базах данных не влияют на внешний интерфейс, так как представления могут скрывать изменения структуры таблиц;
* кодовая реиспользуемость – позволяют сократить код, избавляя от необходимости повторно писать одни и те же запросы.

Представление создается с помощью команды CREATE VIEW.

Рисунок 11 – Создание представления для активных опросов

После создания представления его можно использовать так же, как и таблицу:

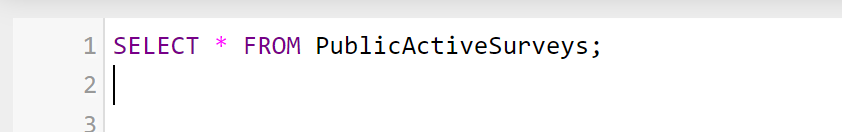


Рисунок 12 – Вывод представления, как таблицу

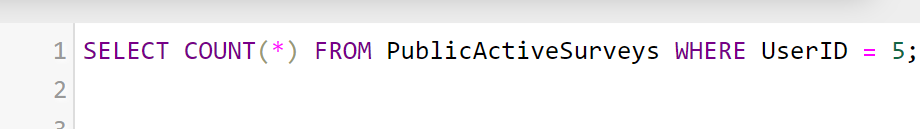


Рисунок 13 – Выведет все активные опросы, сделанные пользователем с ID равным 5

Представления могут быть обновляемыми или не обновляемыми. Обновляемые представления позволяют выполнять операции INSERT, UPDATE и DELETE, при этом изменения будут применяться к базовой таблице.

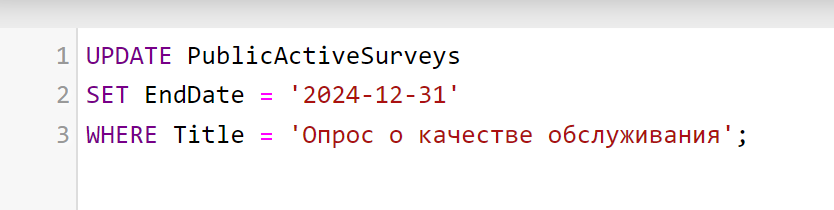
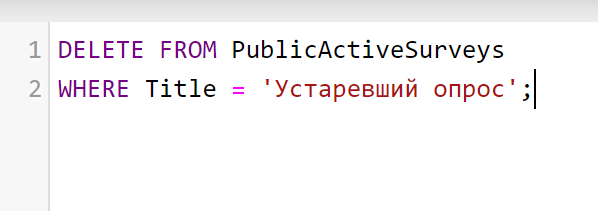


Рисунок 14 – Запрос обновит информацию об окончании опроса в представлении, где титл будет равен “Опрос о качестве обслуживания”

Рисунок 15 – удаление записей через титул

Представления можно изменять с помощью команды CREATE OR REPLACE VIEW или удалять с помощью DROP VIEW

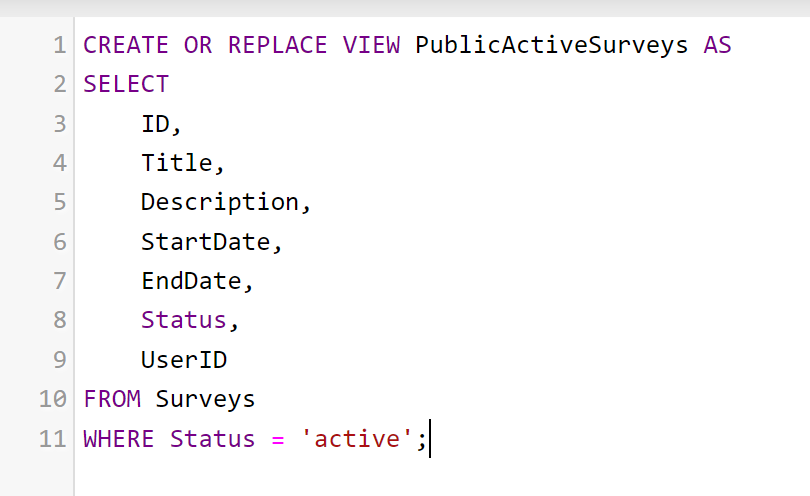


Рисунок 16 – Допустим, вы хотите добавить поле Status в представление, чтобы сделать его обновляемым

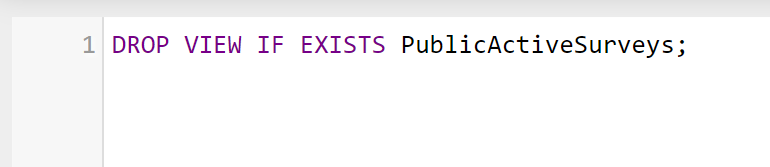


Рисунок 17 – Удаление представления

Представления в базах данных являются мощным инструментом для работы с данными. Они обеспечивают большую гибкость, безопасность и легкость в использовании, способствуя более эффективной организации и обработке информации. Понимание работы с представлениями – важный аспект для разработчиков и администраторов баз данных.

2.5. Процедуры в базе данных

**Хранимые процедуры в базах данных** — это заранее написанные наборы SQL-команд, которые сохраняются на сервере базы данных и выполняются как единая операция. Их можно вызывать по имени, передавая параметры, как функцию в программировании.

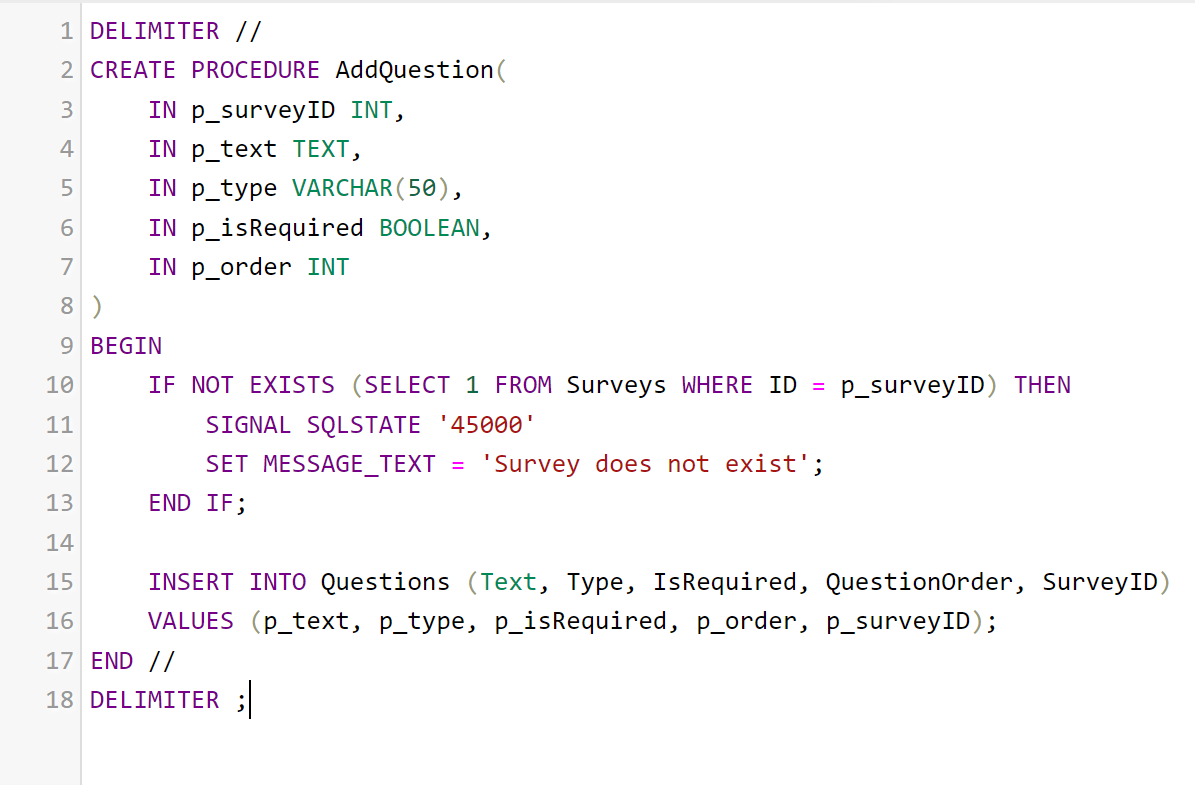


Рисунок 18 – Процедура добавления вопроса к опросу

2.6. Создание ролей в базе данных

Роли в бозе данных — это важный инструмент для управления данными в системе, благодаря ролям можно разделить данные системы для каждой роли, например роли “admin” будет предоставлен инструмент для просмотра жалоб и их скорейшего решения.

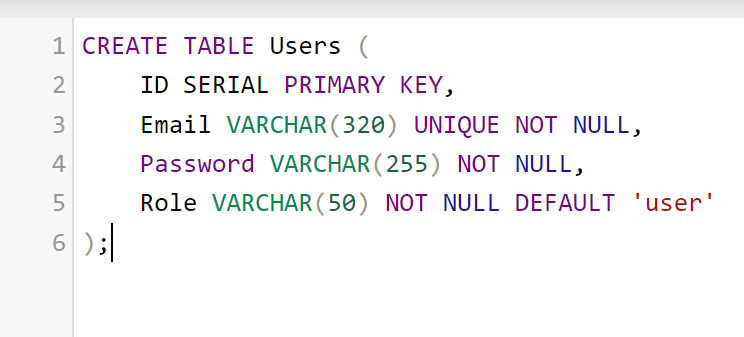


Рисунок 19 – создание таблицы с ролями

Обоснование:

* При обычном создании пользователя ему присваивается роль “user” с помощью модификатора DEFOULT

2.7. Аутентификация пользователей в базе данных

Аутентификация пользователей — это процесс проверки их учетных данных перед предоставлением доступа к системе. В данной курсовой работе реализована система регистрации и входа пользователей, использующая безопасное хранение паролей с хешированием.

**Реализация регистрации пользователя**

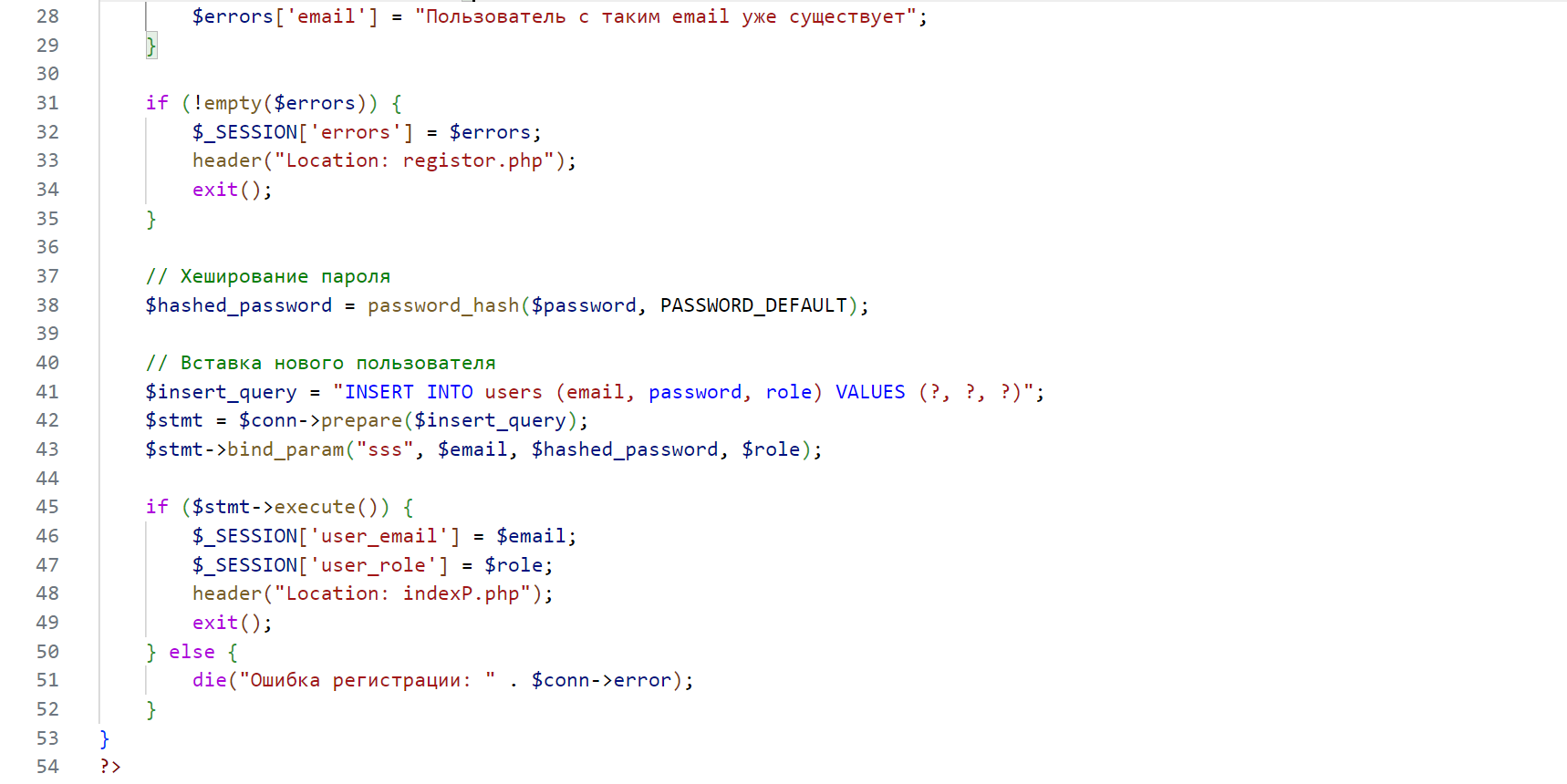


Рисунок 20 – реализация регистрации

Обоснования:

* В начале мы проверяем что данные получены от метода POST.
* Затем все данные передаются в сессию под именем old\_data.
* Идёт инициализация всех переменных из метода POST.
* Проверяется на совпадение повтора пароля, и в случае несовпадения в массив ошибок заносится ошибка.
* Проверяется правильность написания пароля, и в случае неправильного ввода в массив ошибок заносится ошибка.
* Проверяется совпадение почты в уже зарегистрированных данных, если совпадение происходит в массив ошибок заносится ошибка.
* Пароль хэшируется благодаря password\_hash(), хэшируется по алгоритму BCRYPT.
* Подготавливается запрос для sql - инъекции, и если ошибок не произошло, то в сессию сохраняются почта и роль, следом пользователь попадает на главную страницу с возможностью зайти в свой зарегистрированный профиль.

**Реализация входа пользователя**

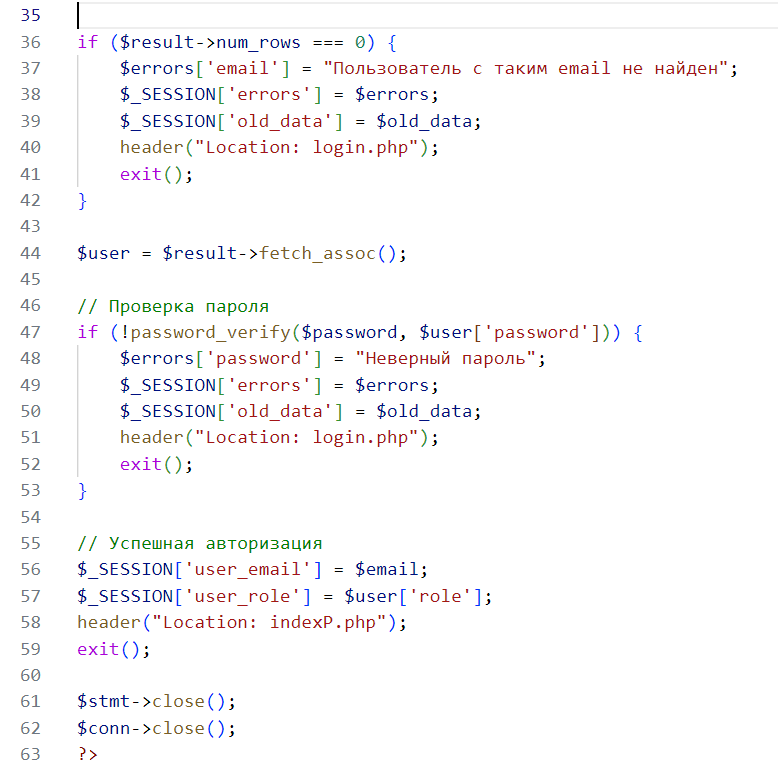
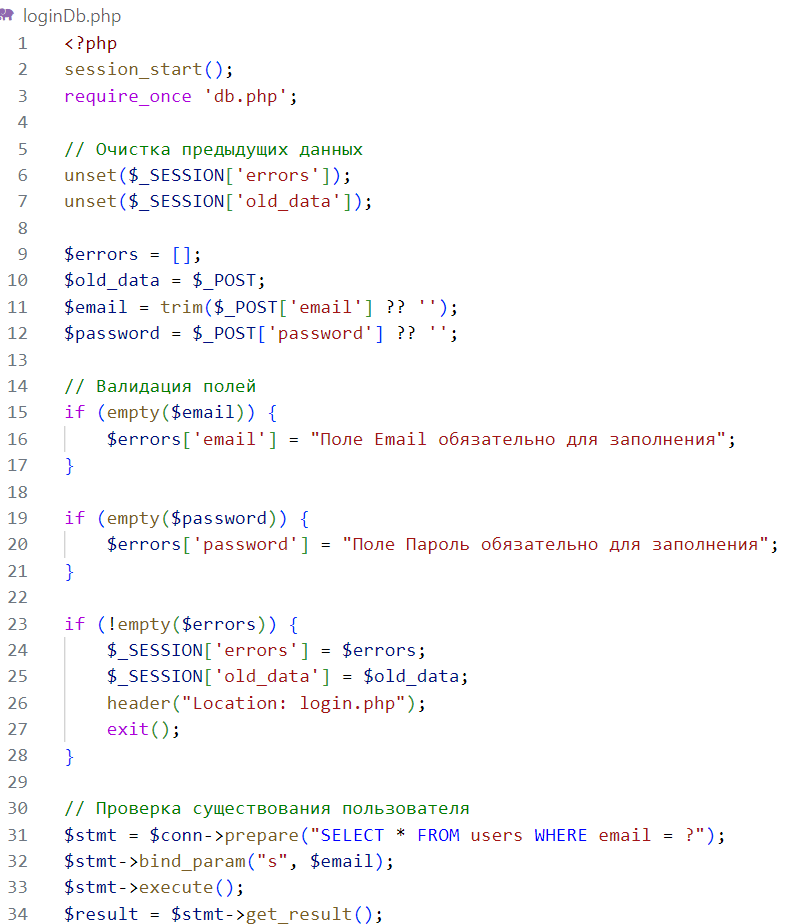


Рисунок 21– реализация авторизации

Обоснования:

* Очищение данных от прошлых ошибок и данных в форме.
* Инициализация переменных.
* Проверка на пустые строки.
* Отправляется запрос в базу данных на нахождение email, если совпадений не нашлось, ошибка записывается в массив ошибок, пользователь переходит на ту же самую форму где был, но уже с упоминанием ошибки
* Если eamil найден, а пароль введён неверно, то ошибка запишится в массив ошибок и сообщит пользователю об непраильном пароле.



Рисунок 22 – подключение к базе данных

Обоснования:

* Инициализируются переменные с данными для подключения к базе данных.
* С помощью mysqli\_connect() в переменную $conn копируется результат подключения.
* Если переменная $conn пустая, сайт умирает от команды die() и пишется ошибка подключения к базе данных.

2.8. Импорт и экспорт базы данных

**Экспорт базы данных**

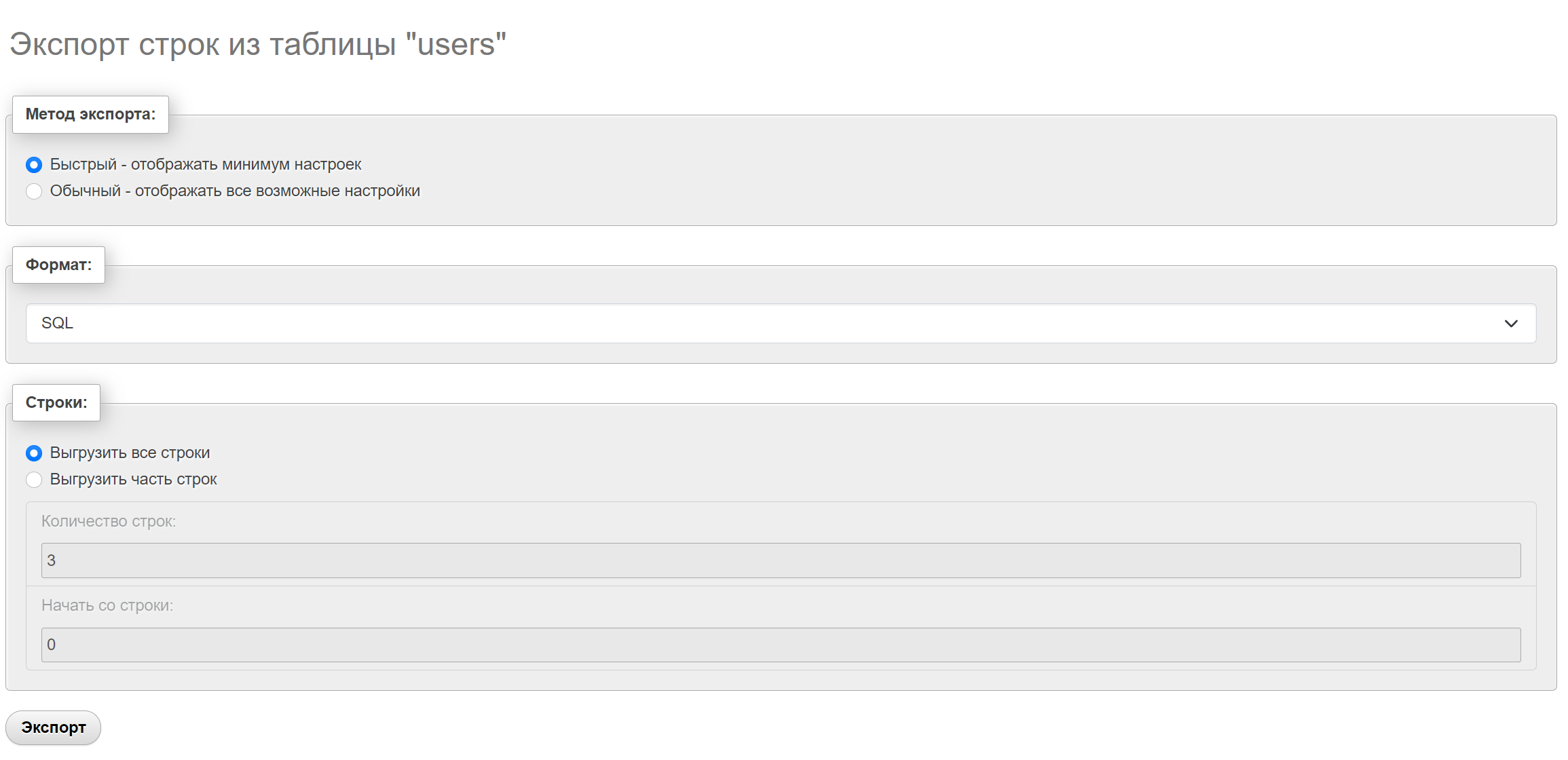


Рисунок 23 – Экспорт базы данных

Для создания резервной копии базы данных выполните следующие шаги:

1. Выбрать базу данных в phpMyAdmin.
2. Перейти во вкладку "Экспорт".
3. Выбрать "Настроенный" метод.
4. В разделе "Таблицы" отметить только нужные таблицы.
5. Нажать "ОК" – будет загружен файл `.sql`, содержащий только выбранные таблицы.

**Импорт базы данных**



Рисунок 24 – Импорт базы данных

Чтобы восстановить базу данных из резервной копии:

1. Открыть phpMyAdmin.
2. Если база данных отсутствует, создать её:

* Перейти во вкладку "Базы данных".
* Ввести имя базы и нажать "Создать".

1. Выбрать созданную базу.
2. Перейти во вкладку "Импорт".
3. Нажать "Выберите файл" и загрузить `.sql` файл.
4. Нажать "ОК", и база данных будет восстановлена.

2.9. Разработка стратегии резервного копирования базы данных

Для разработки стратегии резервного копирования базы данных (БД) сайта онлайн-опросов необходимо учесть критически важные данные, частоту изменений, доступные инструменты и требования к восстановлению.

### **Определение критических данных**

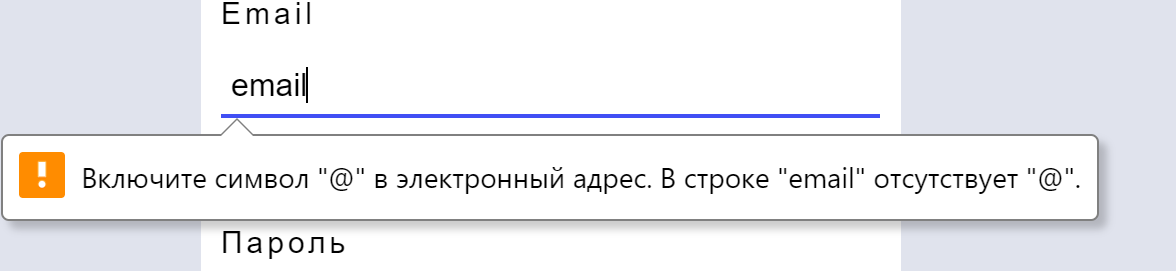
Какие данные нужно резервировать:

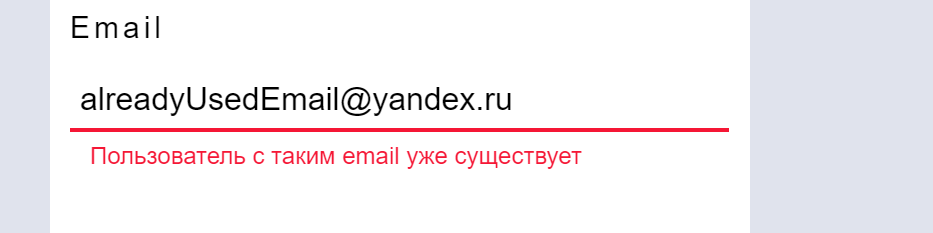
* **Пользователи** (Users): Логины, хэши паролей, роли.
* **Опросы** (Surveys): Названия, описания, даты, статусы.
* **Вопросы** (Questions): Текст, тип, порядок.
* **Ответы** (Response): Данные пользователей, результаты опросов.
* **Жалобы** (Lamentation): Описания, статусы.

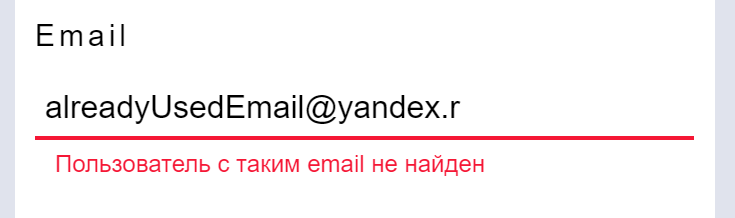
2.10.1. Тестирование базы данных

**Проверка почты**

Проверка почты необходима для защиты от неправильной информации, инструмент проверки служит для предотвращения ошибок связанных с вводом неправильного формата почты, и для устранения регистраций лиц на уже занятые почтовые адреса.

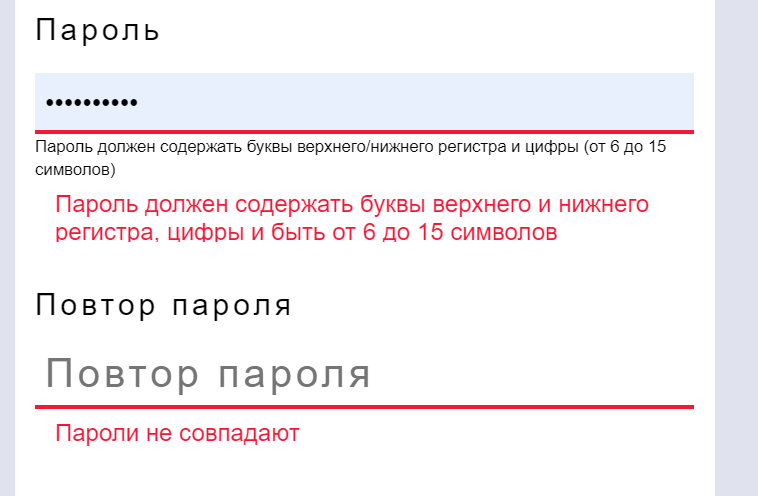
**Регистрация**Рисунок 25 – Неправильный формат почты

Рисунок 26 – Почта уже занята

**Вход**Рисунок 27 – Почта не найдена

**Проверка пароля**

Проверка пароля необходима для защиты пользователя от невнимательности, а так же для уменьшения вероятности взлома аккаунта

**Регистрация**Рисунок 28 – Проверка пароля на правильную валидацию и на повторное написание пароля

**Вход**

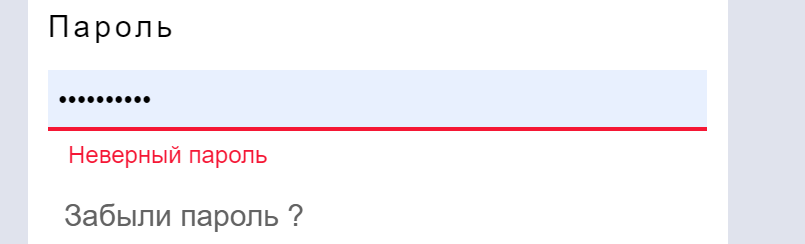
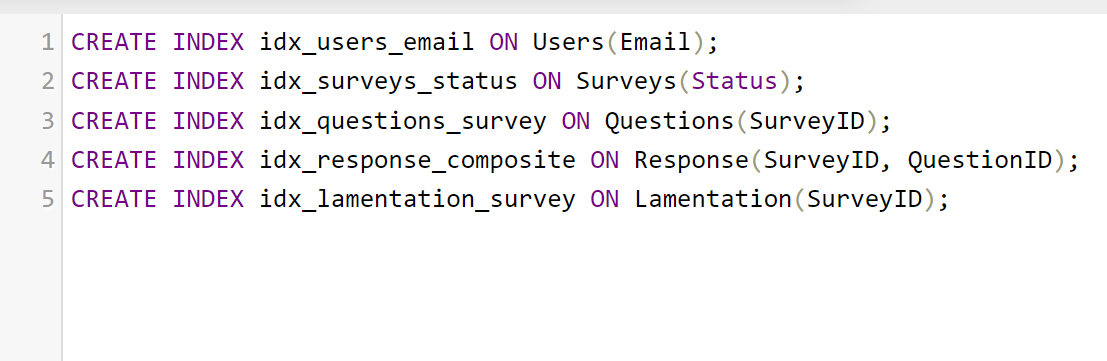


Рисунок 29 – Проверка на правильность пароля привязанного к почте

2.10.2. Оптимизация базы данных

Для оптимизации базы данных используют индексы.

**Индексы в базах данных** — это специальные структуры, которые ускоряют поиск данных в таблицах. Они работают подобно оглавлению в книге: вместо того чтобы перелистывать все страницы (сканировать всю таблицу), вы сразу открываете нужный раздел по указателю.

Рисунок 30 – Проверка на правильность пароля привязанного к почте

ЗАКЛЮЧЕНИЕ

В ходе изучения технологии разработки и защиты баз данных было выявлено, что успешная работа с данными в современных условиях требует комплексного подхода, включающего как аспекты проектирования, так и обеспечения безопасности. Принимая во внимание стремительное развитие информационных технологий, применение современных методов и инструментов становится неизбежным. Это позволяет не только оптимизировать процессы хранения и обработки данных, но и защитить их от возможных угроз.

Особое внимание следует уделить вопросам шифрования данных, аутентификации пользователей и резервного копирования. Эффективная защита баз данных требует активного мониторинга и регулярного обновления систем безопасности.

Таким образом, подходы к разработке и защите баз данных должны быть гибкими и адаптивными, чтобы соответствовать быстро меняющимся требованиям и угрозам. В будущем важно продолжать исследовать и внедрять новые технологии, что позволит обеспечить надежность и безопасность информации на всех уровнях.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

* Волик, М.В. Разработка базы данных в Access / М.В. Волик. – Издательство «Прометей», 2021. – 88 с.
* Даккет, Дж. PHP и MySQL. Серверная веб-разработка / Дж. Даккет. – Издательство «Эксмо», 2023. – 688 с.
* Дьяков, И.А. Базы данных. Язык SQL : учебное пособие / И.А. Дьяков. – Тамбов : Тамбовский государственный технический университет (ТГТУ), 2022. – 82 с.
* Ла Рокка, М. Продвинутые и структуры данных. / М. ла Рокка. – Издательство «Питер», 2024. – 848 с.
* Никсон, Р. Создаём динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 6-е изд. / Р. Никсон. – Издательство «Питер»: Санкт-Петербург, 2023. – 832 с.
* <https://www.figma.com/> (дата обращения: 15.03.2025).
* <https://www.github.com/> (дата обращения: 16.03.2025).

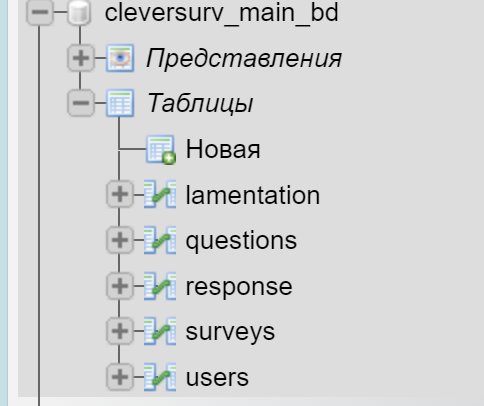
ПРИЛОЖЕНИЯ

Рисунок 31 – Все сущности базы данных

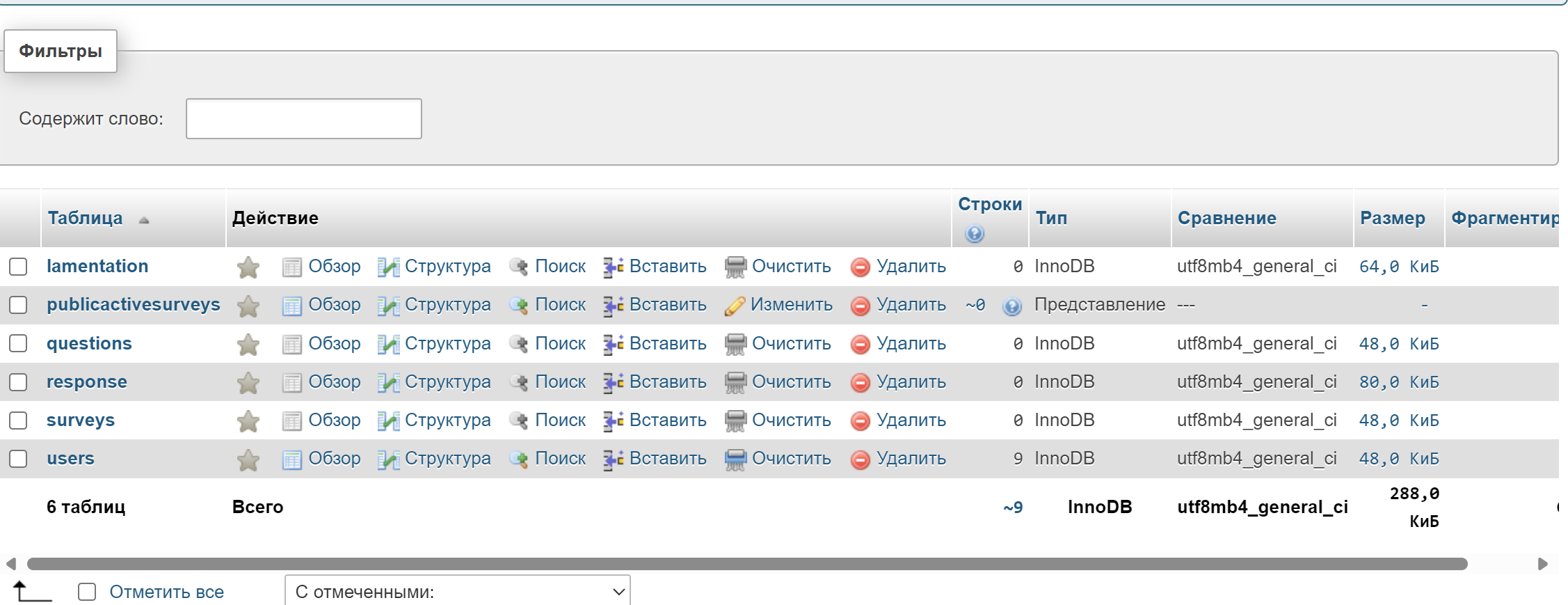


Рисунок 32 – Все сущности базы данных

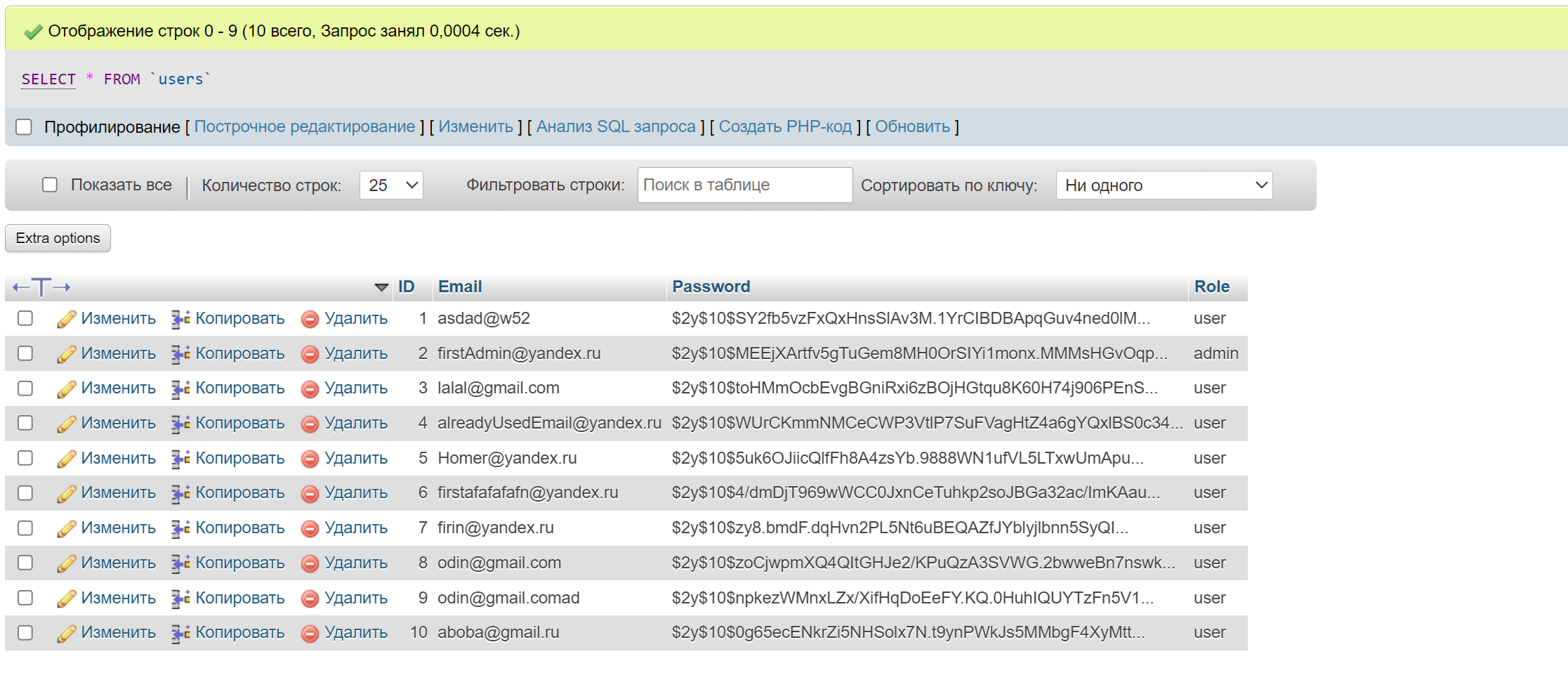


Рисунок 33 – уже добавленные пользователи в таблицу users