МИНИСТЕРСТВО НАУКИ и высшего образования РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования «МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ»

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

филиал «РКТ» МАИ в г. Химки Московской области

**Специальность 09.02.03 «Программирование в компьютерных системах»**

**ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

**ПМ.02«Разработка и администрирование баз данных**»

**Студент**

**Группы МП-32\_\_\_Матросов Евгений Александрович / ( \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ )**

**Руководитель**

**практики от организации \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / ( \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)**

**Руководитель**

**практики от филиала Шумаев А.Ю. / ( \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ )**

**2020г.**

**ПРОГРАММА ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ**

По специальности 09.02.03 «Программирование в компьютерных системах»

|  |  |  |
| --- | --- | --- |
| Дата | Наименование выполняемых работ | Количество дней практики |
| 11.06.20 | Ознакомление с программой практики. Прохождение инструктажа по технике безопасности и охране труда, изучение внутреннего распорядка организации и правил работы. Разработка задания на производственную практику. | 4 |
| 12.06.20 | Описание структуры сети, в которой может функционировать разработанная база данных. | 4 |
| 13.06.20 | Описание сетевого оборудования необходимого для работы с базой данных. | 4 |
| 15.06.20 | Описание программного обеспечения необходимого для работы с базой данных по сети. | 4 |
| 16.06.20-18.06.20 | Разработка клиентских интерфейсов для клиент-серверных приложений. | 16 |
| 20.06.20 | Создание инфокоммуникационной системы.  Выполнение сетевых настроек для взаимодействия с СУБД. | 4 |
| 22.06.20-24.06.20 | Создание концептуальной модели данных.  Создание логической модели данных.  Создание физической модели данных.  Описание механизмов обеспечения целостности базы данных. | 16 |
| 25.06.20-01.07.20 | Описание СУБД представленной базы данных и ее возможностей. Внесение различных данных в базу данных.  Структурирование запросов базы данных.  Определение методов создания хранимых процедур и триггеров. Создание хранимых процедур и триггеров базы данных. | 50 |
| 02.07.20-03.07.20 | Определение способов управления правами пользователей.  Описание распределения прав пользователей и управления ими в базе данных. | 12 |
| 03.07.20-04.07.20 | Определение методов создания и синхронизации реплик базы данных. Описание существующих механизмов репликации в базе данных. Определение методов создания резервных копий базы данных. Описание существующих механизмов резервного копирования в базе данных. | 12 |
| 06.07.20 | Составление отчёта по практике | 6 |
| 07.07.20 | Подготовка к защите отчёта по практике. | 6 |
| 08.07.20 | Итоговая аттестация по производственной практике - зачёт | 6 |

Руководитель практики от филиала «РКТ» МАИ преподаватель

Шумаев А.Ю.

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись)*

**АТТЕСТАЦИОННЫЙ ЛИСТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

студент \_\_\_\_\_\_\_\_\_\_\_Матросов Евгений Александрович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*ФИО*

обучающийся на 3-ем курсе по специальности СПО 09.02.03 «Программирование в компьютерных системах»

успешно прошел производственную практику по профессиональному модулю   
**ПМ.02 «Разработка и администрирование баз данных»**

в объеме 144 часа с 11.06.2020г. по 08.07.2020г.

в организации филиала «РКТ» МАИ

**Виды и качество выполнения работ**

|  |  |  |
| --- | --- | --- |
| Виды работ, выполненных обучающимся(ейся) во время практики | Объем работ | Качество выполнения работ в соответствии с технологией и (или) требованиями организации, в которой проходила практика |
| Ознакомление с программой практики. Прохождение инструктажа по технике безопасности и охране труда, изучение внутреннего распорядка организации и правил работы. Разработка задания на производственную практику. | 4 |  |
| Описание структуры сети, в которой может функционировать разработанная база данных. | 4 |  |
| Описание сетевого оборудования необходимого для работы с базой данных. | 4 |  |
| Описание программного обеспечения необходимого для работы с базой данных по сети. | 4 |  |
| Разработка клиентских интерфейсов для клиент-серверных приложений. | 16 |  |
| Создание инфокоммуникационной системы.  Выполнение сетевых настроек для взаимодействия с СУБД. | 4 |  |
| Создание концептуальной модели данных.  Создание логической модели данных.  Создание физической модели данных.  Описание механизмов обеспечения целостности базы данных. | 16 |  |
| Описание СУБД представленной базы данных и ее возможностей. Внесение различных данных в базу данных.  Структурирование запросов базы данных.  Определение методов создания хранимых процедур и триггеров. Создание хранимых процедур и триггеров базы данных. | 50 |  |
| Определение способов управления правами пользователей.  Описание распределения прав пользователей и управления ими в базе данных. | 12 |  |
| Определение методов создания и синхронизации реплик базы данных. Описание существующих механизмов репликации в базе данных. Определение методов создания резервных копий базы данных. Описание существующих механизмов резервного копирования в базе данных. | 12 |  |
| Составление отчёта по практике | 6 |  |
| Подготовка к защите отчёта по практике. | 6 |  |
| Итоговая аттестация по производственной практике - зачёт | 6 |  |

Руководитель практики от филиала «РКТ» МАИ Шумаев А.Ю.

|  |  |  |
| --- | --- | --- |
| Дата\_\_\_\_\_\_\_\_\_\_\_\_\_ | *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*  *подпись* | *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*  *Расшифровка подписи* |

**ДНЕВНИК** **ПРОХОЖДЕНИЯ ПРАКТИКИ**

|  |  |  |
| --- | --- | --- |
| Дата | Наименование выполняемых работ | Подпись  руководителя |
| 11.06.20 | Ознакомление с программой практики. Прохождение инструктажа по технике безопасности и охране труда, изучение внутреннего распорядка организации и правил работы. Разработка задания на производственную практику. |  |
| 12.06.20 | Описание структуры сети, в которой может функционировать разработанная база данных. |  |
| 13.06.20 | Описание сетевого оборудования необходимого для работы с базой данных. |  |
| 15.06.20 | Описание программного обеспечения необходимого для работы с базой данных по сети. |  |
| 16.06.20-18.06.20 | Разработка клиентских интерфейсов для клиент-серверных приложений. |  |
| 20.06.20 | Создание инфокоммуникационной системы.  Выполнение сетевых настроек для взаимодействия с СУБД. |  |
| 22.06.20-24.06.20 | Создание концептуальной модели данных.  Создание логической модели данных.  Создание физической модели данных.  Описание механизмов обеспечения целостности базы данных. |  |
| 25.06.20-01.07.20 | Описание СУБД представленной базы данных и ее возможностей. Внесение различных данных в базу данных.  Структурирование запросов базы данных.  Определение методов создания хранимых процедур и триггеров. Создание хранимых процедур и триггеров базы данных. |  |
| 02.07.20-03.07.20 | Определение способов управления правами пользователей.  Описание распределения прав пользователей и управления ими в базе данных. |  |
| 03.07.20-04.07.20 | Определение методов создания и синхронизации реплик базы данных. Описание существующих механизмов репликации в базе данных. Определение методов создания резервных копий базы данных. Описание существующих механизмов резервного копирования в базе данных. |  |
| 06.07.20 | Составление отчёта по практике |  |
| 07.07.20 | Подготовка к защите отчёта по практике. |  |
| 08.07.20 | Итоговая аттестация по производственной практике - зачёт |  |

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 7](#_Toc54044900)

[1 Сервер базы данных (SQL-сервер) 8](#_Toc54044901)

[1.1 Функционирование сервера базы данных 8](#_Toc54044902)

[1.2 Преимущества архитектуры клиент-сервер 8](#_Toc54044903)

[2 Программа phpMyAdmin 11](#_Toc54044904)

[2.1 Возможности phpMyAdmin 12](#_Toc54044905)

[2.2 Описание предметной области 22](#_Toc54044906)

[3 Определение состава данных 23](#_Toc54044907)

[3.1 Разработка концептуальной модели 24](#_Toc54044908)

[3.2 Разработка структуры БД 26](#_Toc54044909)

[3.3 Разработка системы управлением контентом 29](#_Toc54044910)

[4 Система отслеживания ошибок 32](#_Toc54044911)

[ЗАКЛЮЧЕНИЕ 36](#_Toc54044912)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 37](#_Toc54044913)

# ВВЕДЕНИЕ

В современном мире, в условиях индустриализации, ежедневно растёт число избыточной информации, в которой иногда трудно сориентироваться и выбрать нужные сведения.

Для решения подобных проблем применяются автоматизированные базы данных. Они стали неотъемлемой частью практически всех компьютерных систем - от отрасли до отдельного предприятия. За последние несколько лет вырос уровень потребительских качеств систем управления базами данных (СУБД): разнообразие поддерживаемых функций, удобный для пользователя интерфейс, сопряжение с программными продуктами, в частности с другими СУБД, возможности для работы в сети и т.д. Серверная часть СУБД позволяет сводить воедино информацию из самых разных источников (электронные таблицы, другие базы данных) и помогает быстро найти необходимую информацию, донести ее до окружающих с помощью отчетов, графиков или таблиц.

Базы данных — это совокупность структур, предназначенных для хранения больших объемов информации и программных модулей, осуществляющих управление данными, их выборку, сортировку и другие подобные действия.

Реляционные базы данных получили широкое применение в различных предметных областях и различных архитектурах.

СУБД Sqlite является программным продуктом с открытым кодом. Обладает рядом преимуществ, таких как быстродействие,компактность,соответствие стандартным требованиям, совместимостью с аналогичными системами.

В качестве серверной части бд, к которой подсоединяются различные клиенты, была выбрана утилита PHPMyAdmin, с помощью которой автоматизирован процесс создания и наполнения таблиц, установки связей в схеме базы данных. Часть процесса разработки выполнен в консольных командах Sqlite.

Приложения и интерфейс базы данных выполнен в скриптах языка PHP.

Практическая работа выполнена в соответствии с рабочей программой ПМ.02«Разработка и администрирование баз данных». Тема данной работы (Магазин игрушек Toy-store). В работе представлены основные этапы анализа и разработки серверной части бд, разработка концептуальной модели базы и логической модели базы данных. В качестве СУБД выбрана система Sqlite архитектуры клиент – сервер.

# 1 Сервер базы данных (SQL-сервер)

СЕРВЕР БАЗЫ ДАННЫХ (англ. database server), также SQL-сервер - сервер, обеспечивающий хранение, обработку и управление структурированными данными (база данных), а также обеспечивающий операции ввода-вывода при доступе клиентов к хранящейся информации. Сервер базы данных занимается обслуживанием базы данных и отвечает за целостность и сохранность хранимых данных. Технически, сервер базы данных представляет собой очень мощный сервер, либо кластер серверов или "ферму серверов", с установленным специальным программным обеспечением, осуществляющим систему управления базами данных (СУБД). Клиенты, которым необходимо прочитать или записать информацию в СУБД, осуществляют доступ к серверу базы данных по сети.

## 1.1 Функционирование сервера базы данных

Основная идея сервера базы данных состоит в том, чтобы размещать, хранить и обрабатывать данные на сервере с мощными вычислительными ресурсами, а с помощью, установленной на нем системы управления базами данных (СУБД) предоставить приложениям, использующим специальные запросы к СУБД, быструю обработку этих запросов и высокоскоростной доступ. Для организации работы такой системы применяется архитектура клиент-сервер. Для управления современными базами данных как правило используется язык структурированных запросов SQL (англ. structured query language). Сервер, работающий под управлением СУБД, использующей язык SQL, называется SQL-сервер. В настоящее время наиболее распространены следующие программные продукты (SQL-серверы), реализующие систему управления базами данных (СУБД): Microsoft SQL Server; Oracle Database Server; IBM DB2; MySQL, PostgreSQL и другие.

## 1.2 Преимущества архитектуры клиент-сервер

Информационные системы, использующие архитектуру «клиент-сервер», обладают серьезными преимуществами по сравнению с их аналогами, созданными на основе сетевых версий настольных СУБД.  
Одним из важнейших преимуществ архитектуры «клиент-сервер» является снижение сетевого трафика при выполнении запросов. Hапример, при необходимости выбора пяти записей из таблицы, содержащей миллион записей, клиентское приложение посылает серверу запрос, который сервером анализируется на корректность и, если запрос корректен, компилируется, оптимизируется и выполняется. После этого результат запроса (те самые пять записей, а вовсе не вся таблица) передается обратно клиенту. При этом, формулируя запрос, можно не задумываться о том, есть ли в базе данных индексы, способные облегчить поиск нужных записей, — если они есть, то они будут использованы сервером, а если нет,запрос все равно будет выполнен, хотя, возможно, это займет больше времени, чем при использовании индексов. Но в любом случае — есть индексы или нет — в клиентское приложение передается только результат запроса, и в этом случае сетевой трафик не зависит ни от их наличия, ни от числа записей в таблицах, к которым произведен запрос.  
Вторым преимуществом архитектуры «клиент-сервер» является возможность хранения бизнес-правил (например, правил ссылочной целостности или ограничений на значения данных) на сервере, что позволяет избежать дублирования кода в различных клиентских приложениях, использующих общую базу данных. Кроме того, в этом случае любое редактирование данных, в том числе и редактирование средствами, не предусмотренными разработчиками информационной системы (например, различными утилитами администрирования сервера), может быть произведено только с учетом этих правил. Если последние версии некоторых настольных СУБД и способны хранить ограничения на значения данных либо тексты SQL-запросов, триггеры и хранимые процедуры в них, как правило, отсутствуют — их просто некому выполнять.   
Третье преимущество:Для описания серверных бизнес-правил в наиболее типичных ситуациях (например, при реализации связей master/detail) нередко используются так называемые CASE-средства (CASE означает Computer-Aided System Engineering) для создания диаграмм «сущность-связь». Эти инструменты позволяют описать подобные правила на уровне логической и физической моделей данных без какого бы то ни было программирования, а затем сгенерировать код соответствующих серверных объектов — триггеров, хранимых процедур, серверных ограничений. В этом случае клиентские приложения будут избавлены от значительной части кода, связанного с реализацией бизнес-правил непосредственно в приложении. Можно Отметить также, что часть кода, связанного с обработкой данных, также может быть реализована в виде хранимых процедур сервера, что позволяет еще больше «облегчить» клиентское приложение, а это означает, что требования к рабочим станциям могут быть не столь высоки. Это в конечном итоге снижает стоимость информационной системы даже при использовании дорогостоящих серверной СУБД и аппаратного обеспечения.  
  
Помимо перечисленных выше трех преимуществ следует также отметить, что современные серверные СУБД обладают широкими возможностями управления пользовательскими привилегиями и правами доступа к различным объектам базы данных. Как правило, в базе данных хранятся сведения о ее пользователях, их паролях и привилегиях, а каждый объект базы данных принадлежит какому-либо пользователю. Владелец объекта может предоставить другим пользователям право использовать его тем или иным способом (например, позволить читать из него данные какому-либо другому пользователю). Большинство серверных БД поддерживает так называемые роли, представляющие собой совокупность прав на доступ к тем или иным объектам базы данных. В этом случае каждый пользователь может иметь одну или несколько ролей и соответственно определенные в этих ролях привилегии.  
Современные серверные БД обладают также широкими возможностями резервного копирования и архивации данных, а нередко и оптимизации выполнения запросов. Они также, как правило, предоставляют возможность параллельной обработки данных, особенно в случае использования многопроцессорных компьютеров в качестве сервера баз данных.

2 Программа phpMyAdmin

Программа phpMyAdmin является одним из самых распространенных средств для администрирования, управления и каждодневной работы с базами данных Sqlite. Практически на каждом хостинге, где установлена поддержка этой СУБД, в качестве панели управления предлагают phpMyAdmin. Что же собой представляет этот инструмент?

phpMyAdmin – это набор скриптов, написанных на РНР, который предоставляет практически все необходимые функции по работе с базами данных Sqlite. В дополнение к возможностям самого сервера Sqlite, он предоставляет дополнительные "фичи", которые позволяют более эффективно и легко работать с данными. Причем все функции доступны прямо из браузера, даже перезагрузка удаленного сервера (если эта возможность разрешена учетной записью пользователя).

Толчком послужил выпуск новой версии программы. Кардинально новых возможностей уже давно не появляется – ведь все необходимое уже давно реализовано. Но теперь разработчики наконец принялись за "облагораживание" своего продукта и переделали интерфейс. Раньше он было довольно унылый и полностью текстовый, потом постепенно начали добавляться пиктограммы и улучшаться внешний вид. 2.6.0-pl3 Наконец получил качественное оформление, и хотя сама структура и принципы работы остались неизменными, именно как инструмент программа стала намного более приятной. На рисунке 1 представлен вид интерфейса phpMyAdmin.

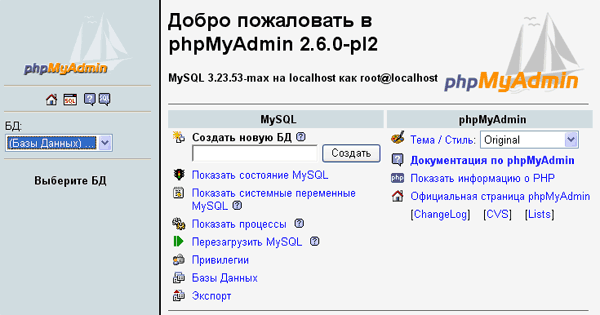


Рисунок 1 –Интерфейс phpMyAdmin

phpMyAdmin нужен для администраторов серверов баз данных на хостинговых площадках. Администраторы могут легко создавать аккаунты для пользователей, назначать им привилегии, создавать и настраивать базы данных. Причем, и пользователи, и администраторы могут использовать один и тот же скрипт – просто у пользователя, не обладающего необходимым уровнем доступа, не будет некоторых функций – например, выбора сервера БД, доступа к служебным базам и базам других пользователей и т.д. Но вся функциональность в рамках созданной для него базы полностью доступна.

Разработчики получают полный контроль над своим сервером, над всеми базами и таблицами, возможность интерактивно исполнять SQL-запросы и даже начальные функции отладки неверных запросов.

## 2.1 Возможности phpMyAdmin

phpMyAdmin имеет множество возможностей которые представлены в главном меню.Главное Меню phpMyAdmin имеет много пунктов. Главное меню представлено представлено на рисунке 2

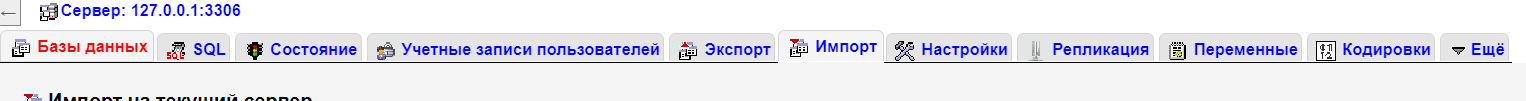


Рисунок 2 –Главое меню phpMyAdmin

Пункты Главного меню phpMyAdmin:

* Пункт «Базы данных» — открывает список всех баз данных, доступных для текущего пользователя phpMyAdmin и ссылки на них.
* Пункт «SQL» — открывает окно в котором можно выполнить SQL-запрос(ы) на сервере.
* Пункт «Состояние» — выводит статистику работы SQL-сервера и инструменты для работы с ним.
* Пункт «учётные записи пользователей» — открывает список всех пользователей баз данных, доступных для текущего пользователя phpMyAdmin, ссылки на них и другие инструменты для работы с пользователями баз данных на сервере.
* Пункт «Импорт» — открывает окно для импорта баз данных на текущий сервер и его настройками.
* Пункт «Экспорт» — открывает окно для экспорта баз данных с текущего сервера и его настройками.
* Пункт «Настройки» — большой пункт меню, который содержит многочисленные настройки программы phpMyAdmin.
* Пункт «Репликация» — открывает окно для работы по репликации сервера баз данных и настройками репликации.
* Пункт «Переменные» — открывает окно со списком серверных переменных и возможностью их настройки.
* Пункт «Кодировки» — справочная информация в виде списка доступных кодировок таблиц баз данных.
* Пункт «Типы таблиц» — справочная информация по типам таблиц баз данных SQL и их характеристиками. Интерфейсы и дополнительное описание пунктов представлены на рисунках 3-17

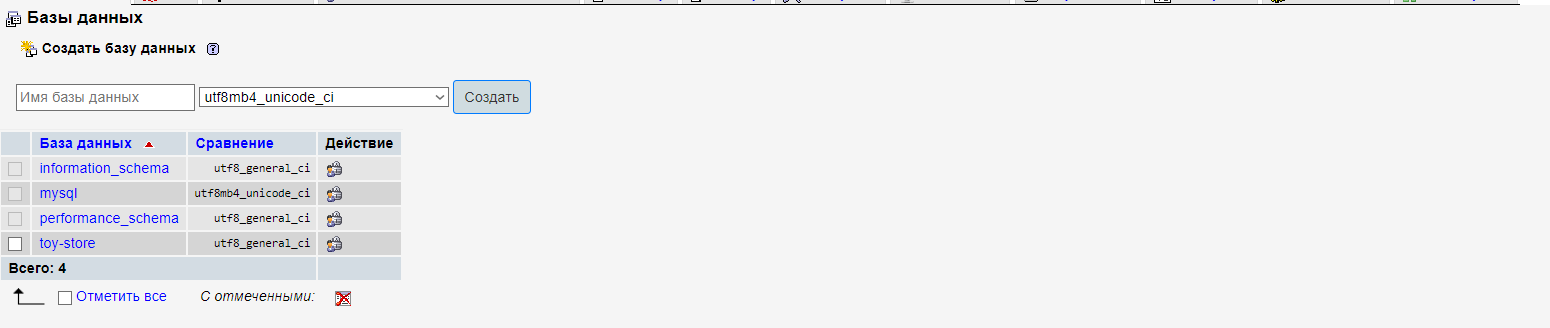
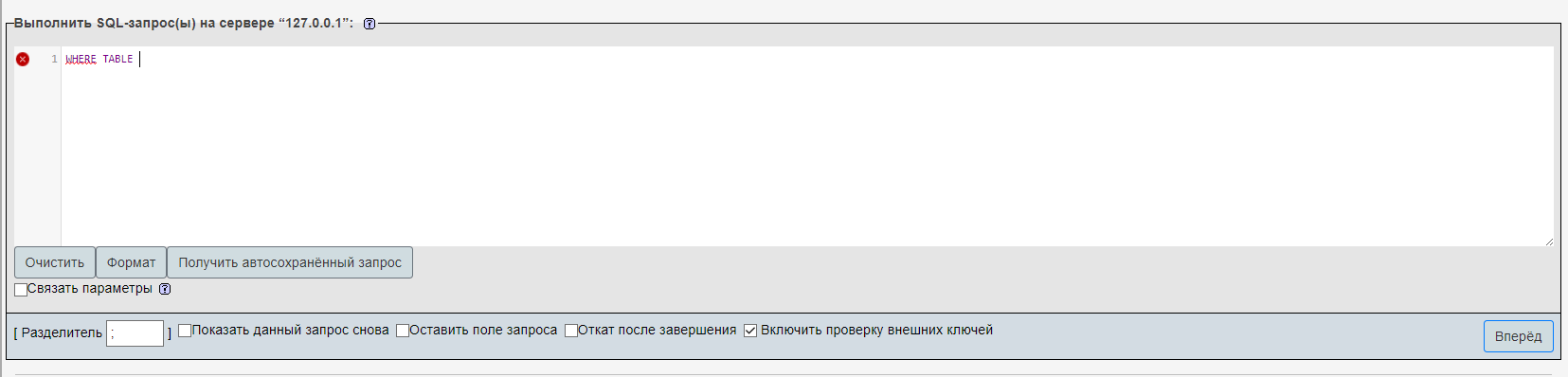


Рисунок 3 –список базы данных

 Рисунок 4 –окно выполнения sql запросов на сервере

Пункт состояние представлен на рисунке 5. Также пункт состояние имеет подпункты:

* Сервер- отображает информацию о трафике, соединении,когда запущен сервер
* Процессы-таблица которая содержит информацию о пользователе,хосте,база данных,команда,время,состояние,прогресс,sql запрос.
* Статистика запросов- содержит параметры и диаграмму к ним
* Все переменные состояния- список переменных со значениями и описанием, также имеются фильтры.
* Монитор- имеет графический мониторниг параметров таких как загрузка процесса,системная память,система подкачки и тд.
* Советчик- в нём описаные основные проблемы и рекомендации для их решения.



Рисунок 5 –пункт состояние

В списке учётных записей имеется информация о имени пользователя, имени хоста,имеется ли пароль, привилегии, также справа имеется графа действий и редактирования. Список учётных записей пользователей представлен на 6.

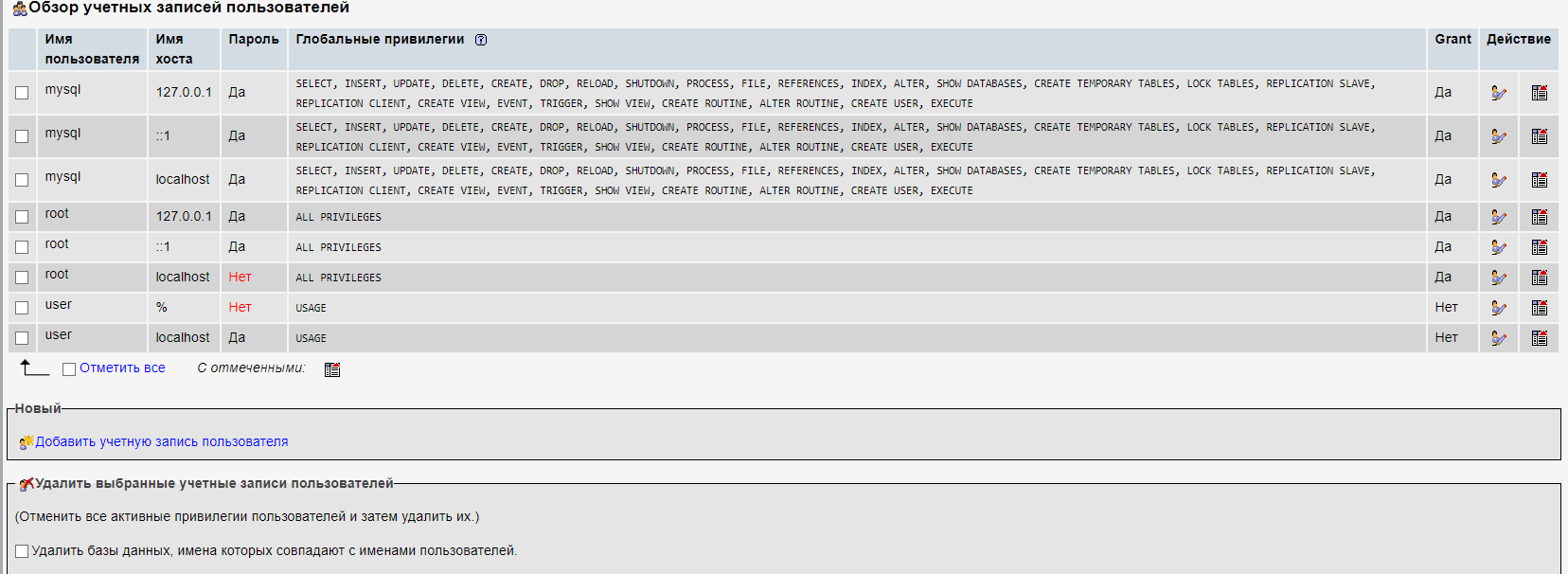


Рисунок 6 –список учётных записей пользователей

Для того чтобы сделать импорт базы данных на хостинг с веб-интерфейсом phpMyAdmin нам понадобится пункт импорт. Процедура импорта базы данных представлена на рисунке 7.

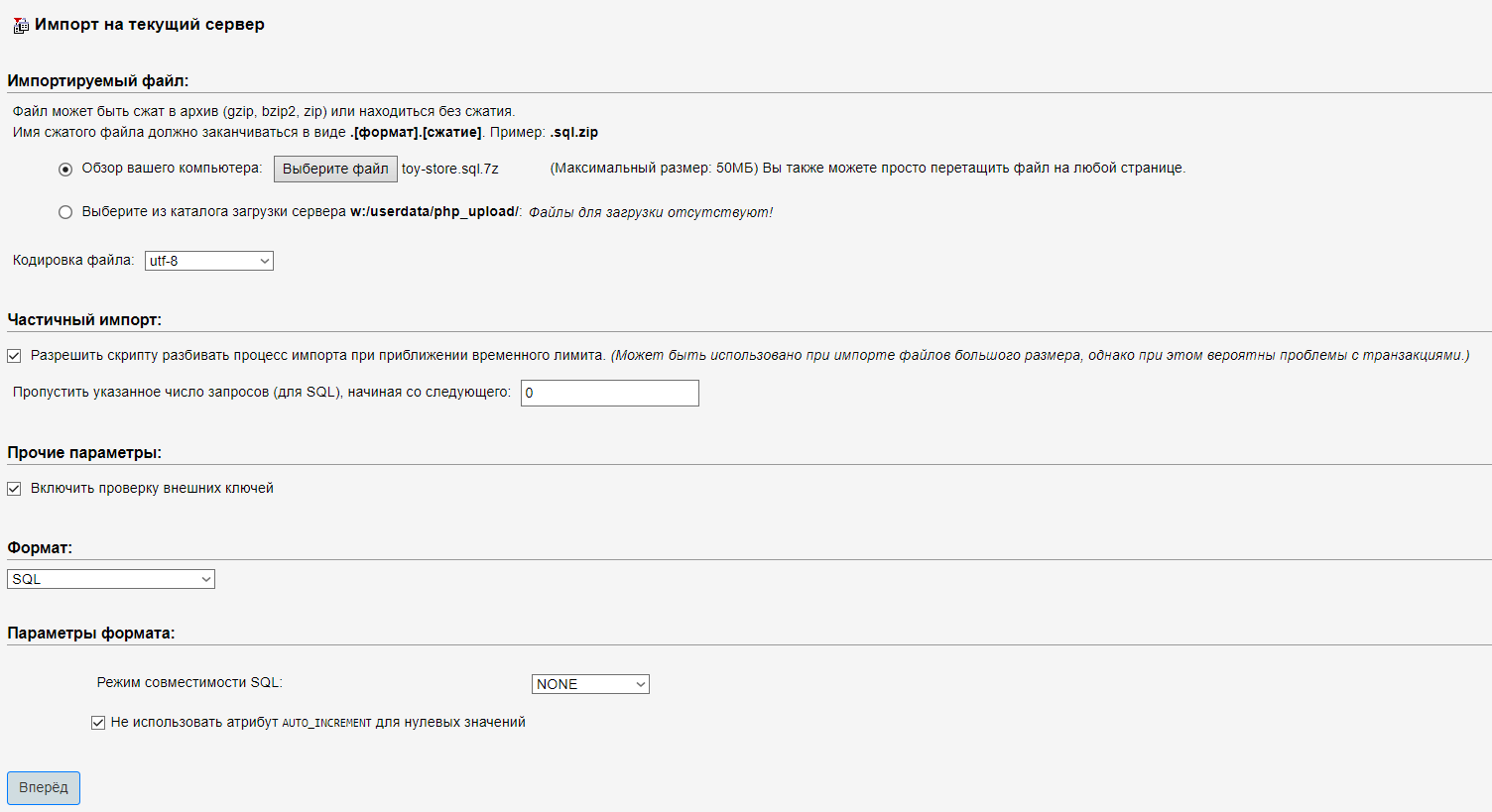


Рисунок 7 –Импорт базы данных на сервер

Также можно сделать экспорт базы данных к себе на компьютер, для этого есть два метода экспорта это:

1. Быстрый
2. Обычный

На рисунке 8 представлена процедура экспорта базы данных

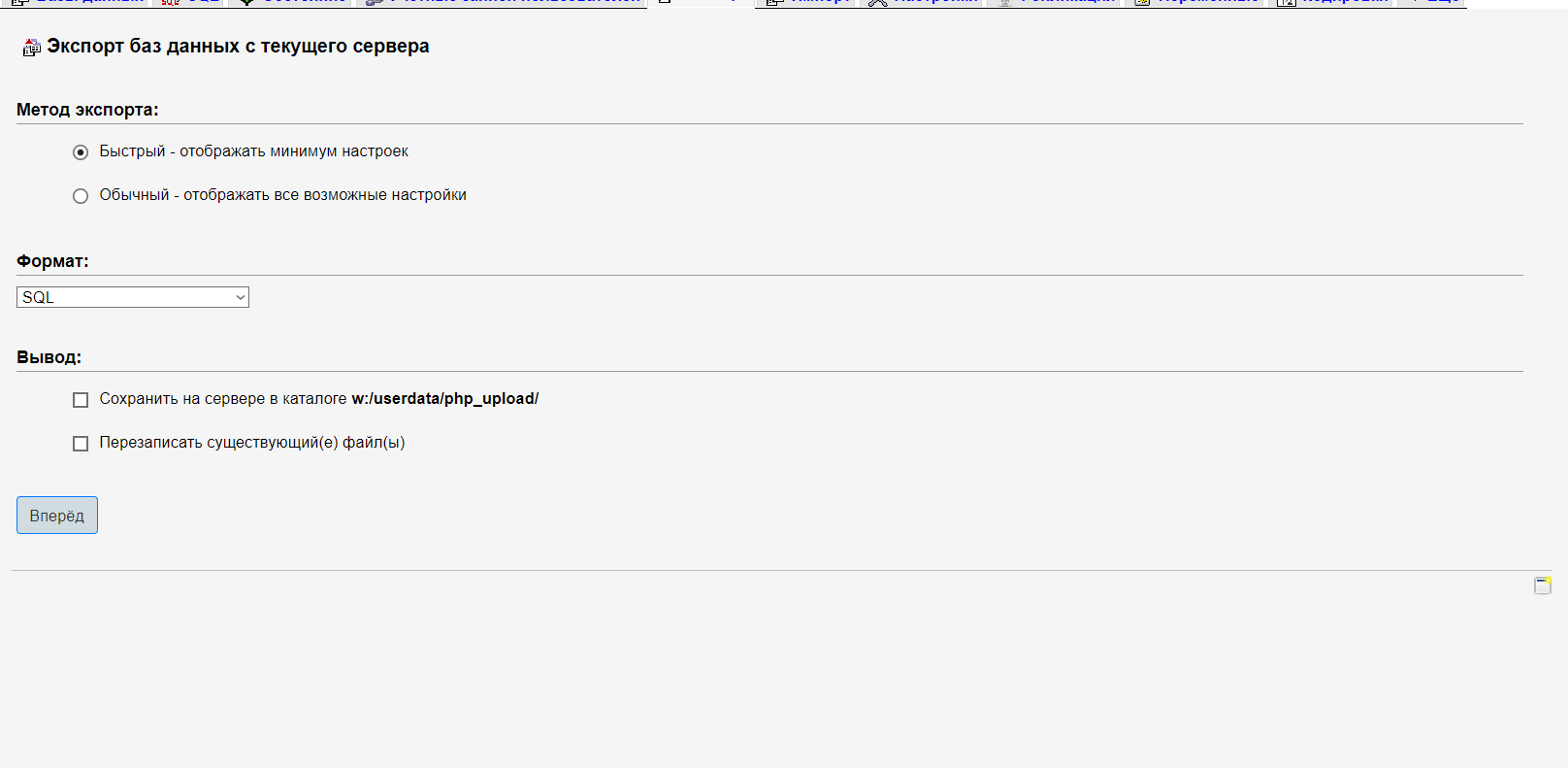


Рисунок 8 –Экспорт базы данных на сервер

1. Быстрый способ экспорта базы данных подходит,если сайт и бд сделана только что и база данных почти пустая. В этом случае, если сделать экспорт базы данных, при помощи быстрого экспорта, то импорт пройдет без ошибок.
2. Обычный способ экспорта подходит,если осуществляется перенос сайта на другой хостинг или домен,либо делается резервная копия то, если вы воспользоваться быстрым экспортом, при импорте будут возникать ошибки. Для того, что бы это избежать, нужно использовать обычный экспорт с определёнными настройками. На рисунках 9-12 представлены параметры настройки обычного метода экспорта.

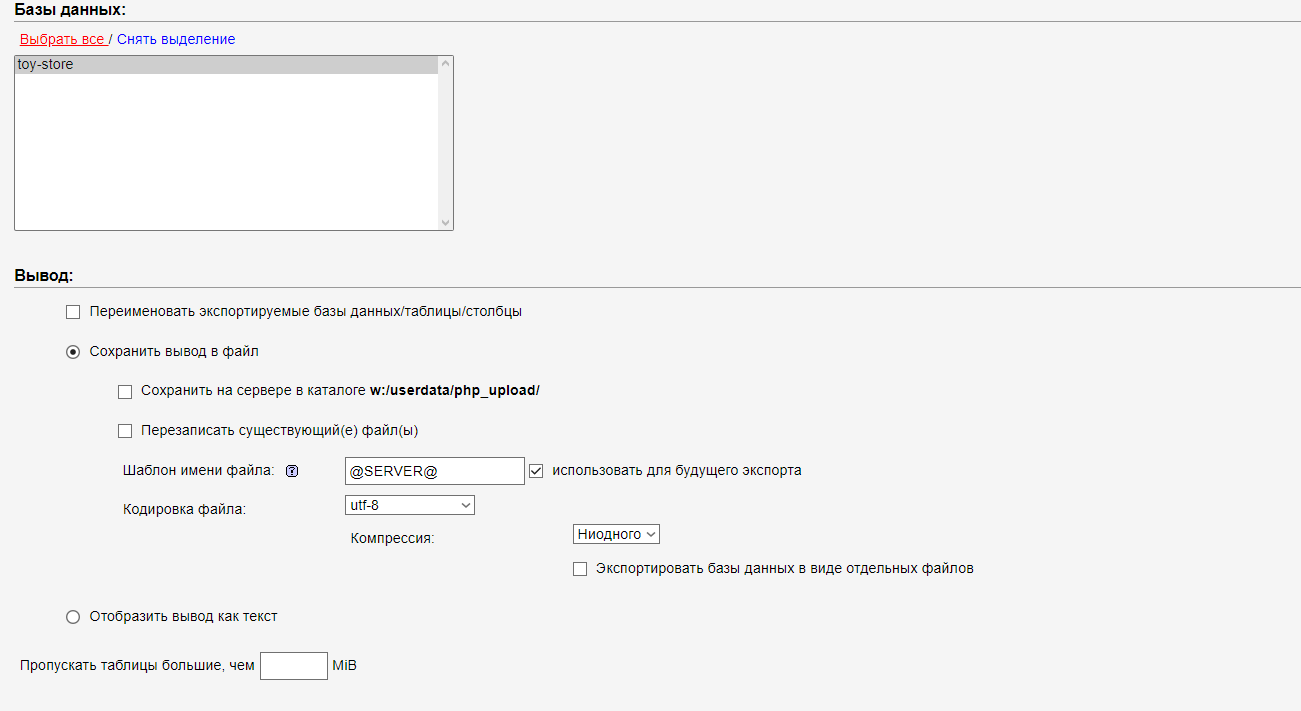


Рисунок 9 –Параметры настройки обычного метода экспорт

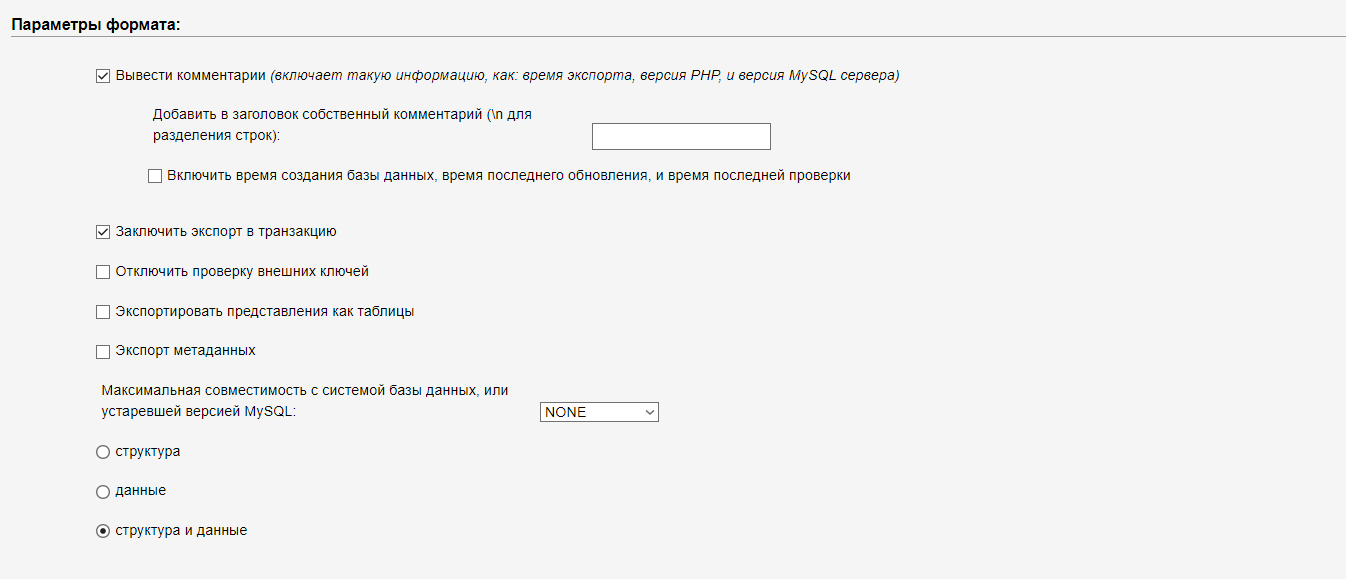


Рисунок 10 –Параметры настройки обычного метода экспорта

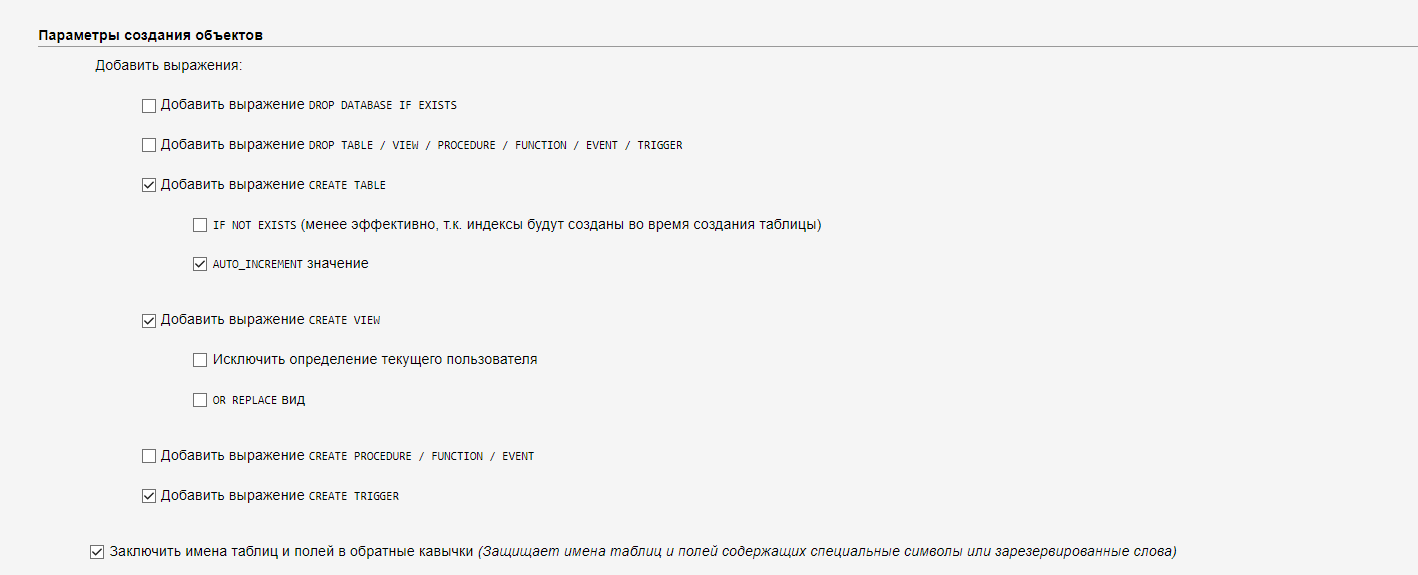


Рисунок 11 –Параметры настройки обычного метода экспорта

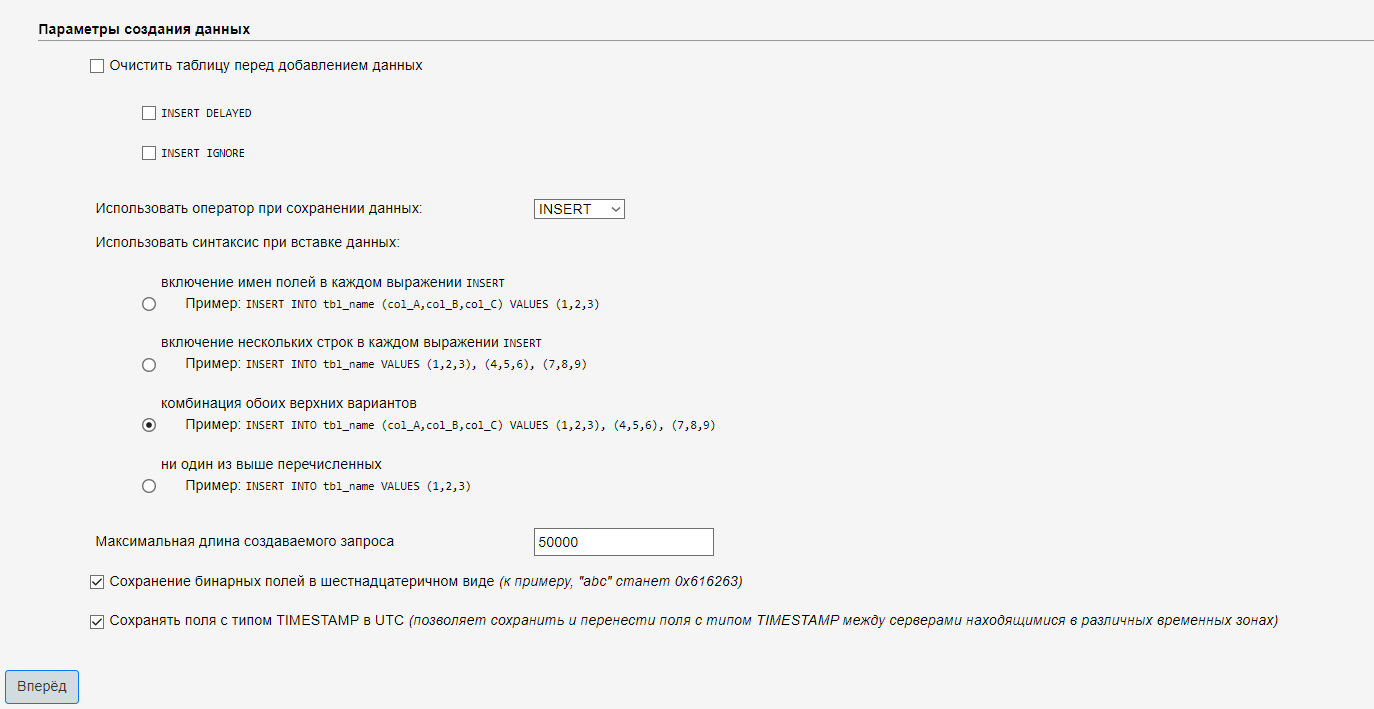


Рисунок 12 –Параметры настройки обычного метода экспорта

PhpMyAdmin имеет окно основных настроек которое отвечает за работу основных возможностей.

Окно настроек представлено на рисунке 13, также Окно настроек имеет подпункты:

* Пользовательский настройки- в которых пользователь может сделать импорт и экспорт файлов и сбросить настройки
* Двухфакторная аунтефикация- дополнительный уровень безопасности который имеет настройку
* Настройки-имеет общие настройки такие как проверка версии,порядок сортировки,исходное положение раскрывающихся блоков, отключение обслуживания таблиц,отключение сочетания клавиш, отправка отчётов об ошибка. Настройки базы данных,текстовых полей, заголовков страниц с параметрами:сортировки, максимального и минимального количества (баз данных,заголовков),ширина и высота texarea, количество столбцов в текстовом поле и тд.
* Sql запросы- в нём есть такие настройки как: длина истории запросов,проверка внешних ключей, показывать запросы и подтверждать, максимальная длина отображаемого и тд.
* Панель навигации- имеет настройки дерева навигации,сервера,базы данных,таблицы.
* Основная панель- включает в себя настройки структуры базы данных,структуру таблиц,вкладки, обзоры,отображение схемы связей.
* Настройки экспорта и импорта

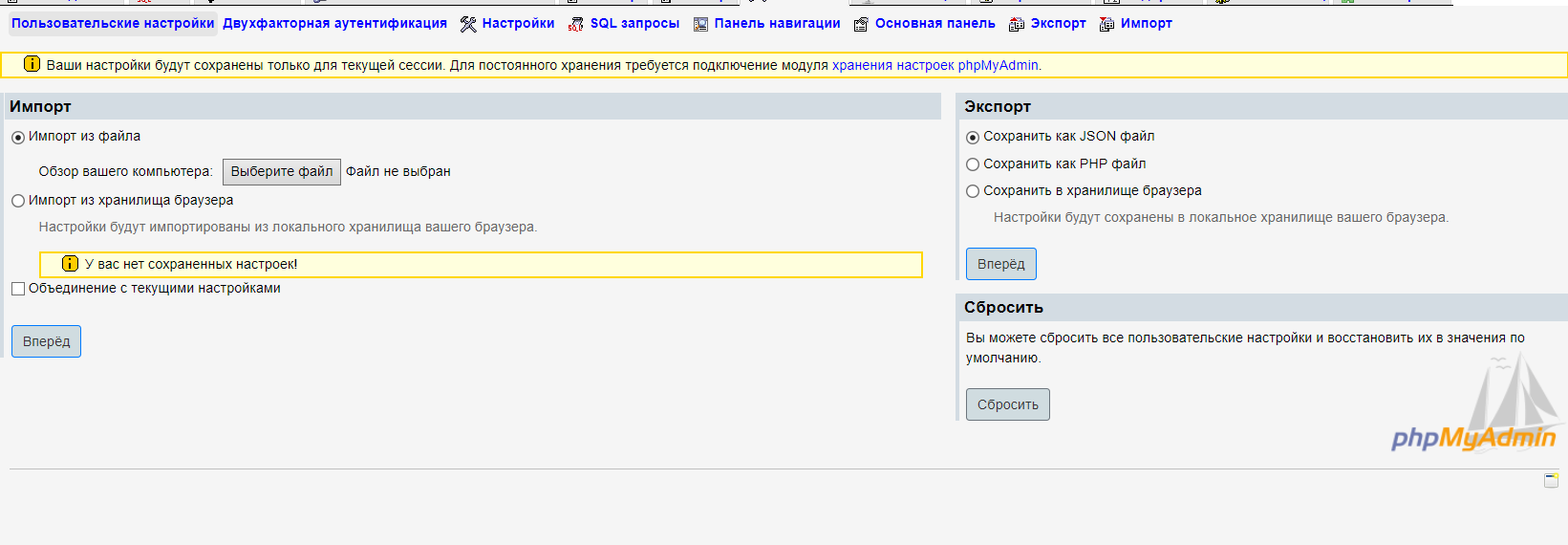


Рисунок 13 –окно настроек программы phpMyAdmin

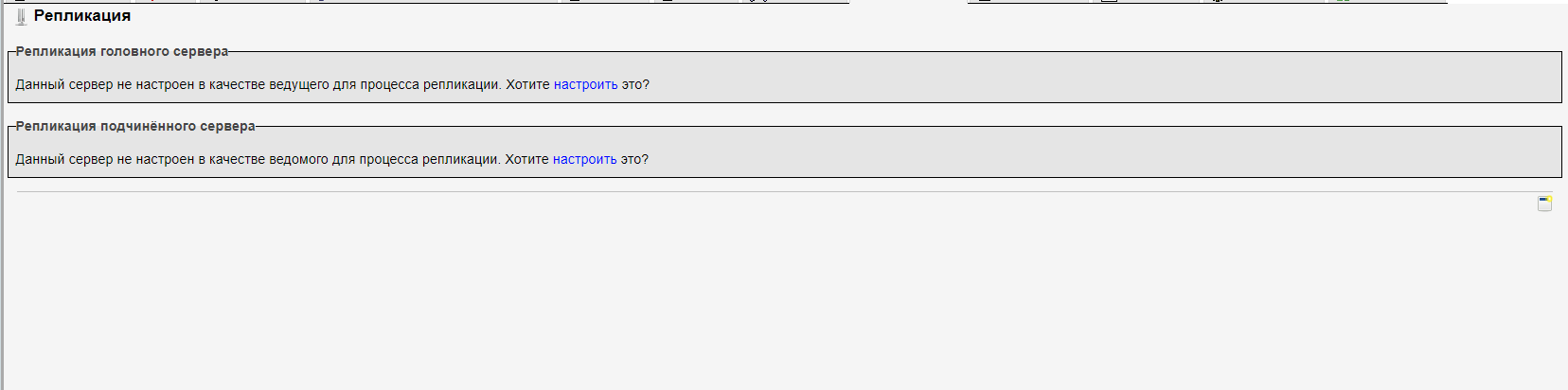


Рисунок 14 – окно для работы по репликации сервера баз данных и настройками репликации

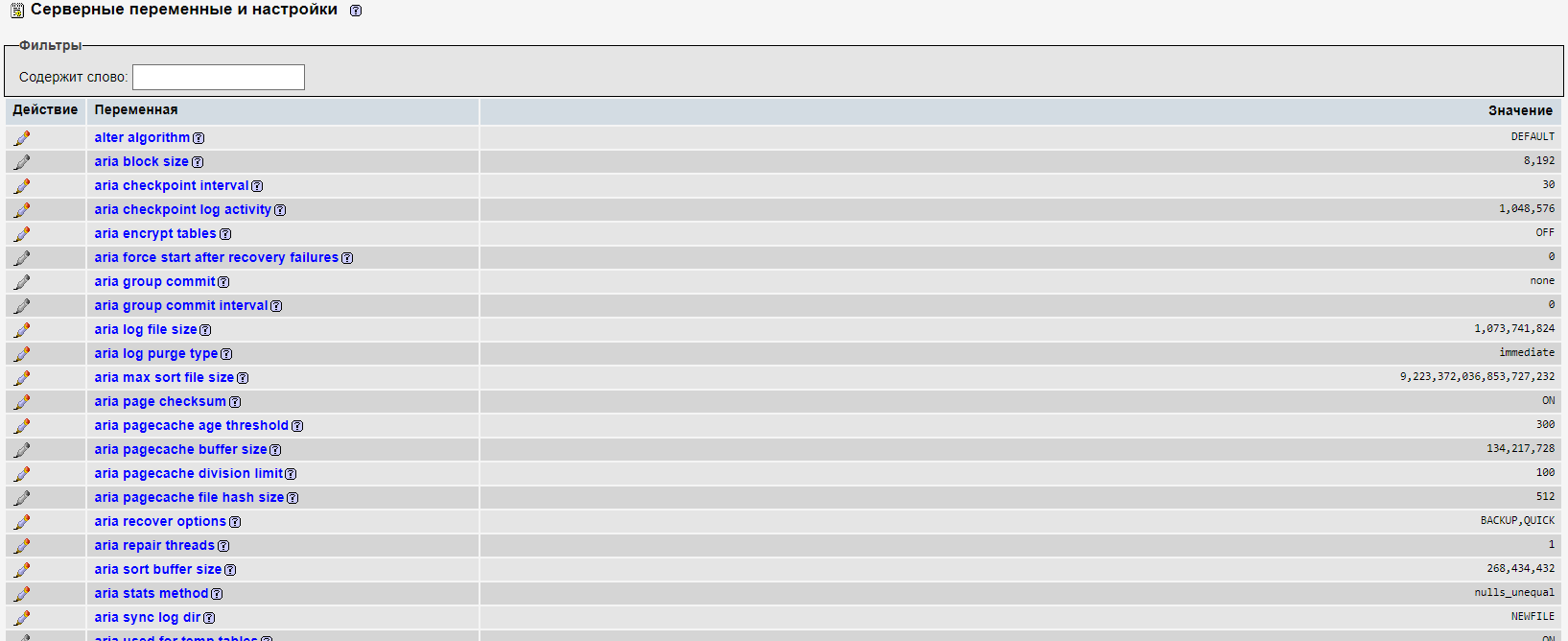


Рисунок 15 –список серверных переменных

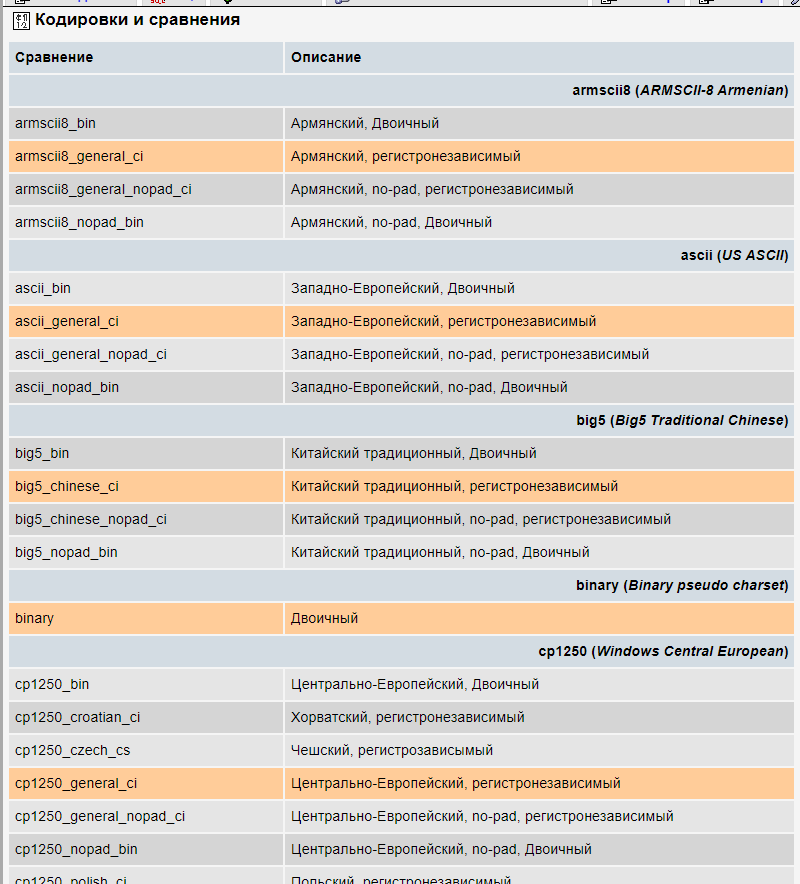


Рисунок 16 –  список доступных кодировок таблиц баз данных

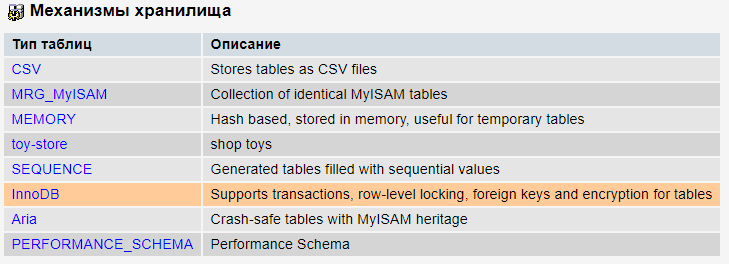


Рисунок 17 – информация по типам таблиц баз данных  и их характеристиками

phpMyAdmin имеет встроенные средства для проведения текущих работ с базами и таблицами – обслуживание, проверка и починка испорченных таблиц, резервирование и восстановление из архивов, экспорт данных из таблиц, как в формате SQL, так и в более специфических, например, LaTeX или PDF. Для уменьшения объема файл с данными можно сразу же на сервере сжать архиватором - это очень полезно для создания на своей машине копии из удаленной базы на сайте. Структура таблиц показана на рисунке 18.

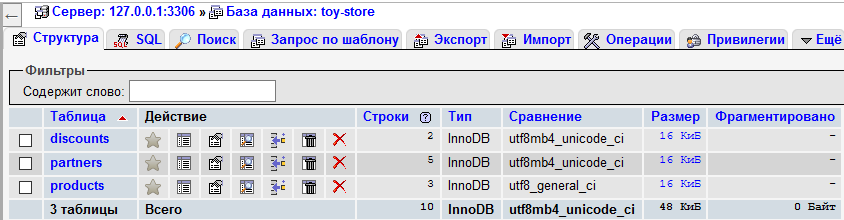


Рисунок 18 - Структура таблиц базы данных toy-store

## 2.2 Описание предметной области

Магазин игрушек Toy-store занимается продажей игрушек в ассортименте. Предположим, что для данной цели была разработана база данных, содержащая в себе информацию о каждой игрушке. Для разработки данной базы данных необходимо обладать следующей информацией:

* Название игрушки
* Цена игрушки
* Описание игрушки
* Картинка товара
* Оценка игрушки
* Количество просмотров

Администратор в магазине игрушек Toy-store должен вводить данные в виде записей базы, таких как:

* Код товара
* Название игрушки
* Цена игрушки
* Описание игрушки
* Картинка товара(игрушки)
* Оценка игрушки

Информационная система при запросе от пользователя, в данном случае, Администратора предоставляет информацию о товаре(игрушке) в полной мере, в которой она содержится в базе данных.

3 Определение состава данных

В разрабатываемой базе данных имеется информация о каждой конкретной игрушке, что делает возможным обеспечение пользователя данной информацией. База данных представляет собой систему, состоящую из взаимосвязанных таблиц и связей между ними. В результате анализа предметной области структура содержания таблиц в БД определяется следующим образом: БД содержит в себе такие таблицы, как: «discounts», «partners», «products».

На основе анализа предметной области можно выявить логические взаимосвязи данных, которые определяют структуру данных предметной области. В результате должны быть выявлены информационные объекты и установлены связи между ними. Физическая модель – логическая модель базы данных, выраженная в терминах языка описания данных конкретной БД.

На рисунке 19 представлена физическая модель базы данных со связями между таблицами

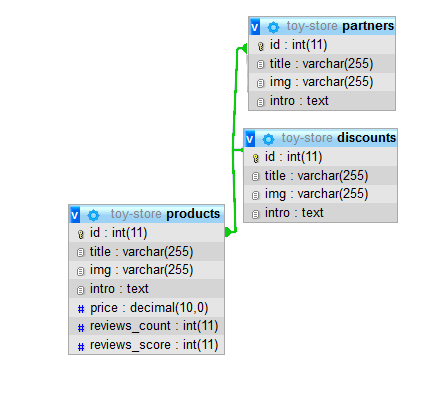


Рисунок 19 - Связи таблиц базы данных «Магазин игрушек Toy-store»

3.1 Разработка концептуальной модели

Концептуальная модель – модель предметной области, состоящей из перечня связанных понятий, используемых для описания области, вместе со свойствами и характеристиками, классификацией этих понятий, по видам, ситуациям, признакам в данной области и алгоритмов протекания процессов в ней. На рисунке 20 представлена концептуальная модель базы данных.

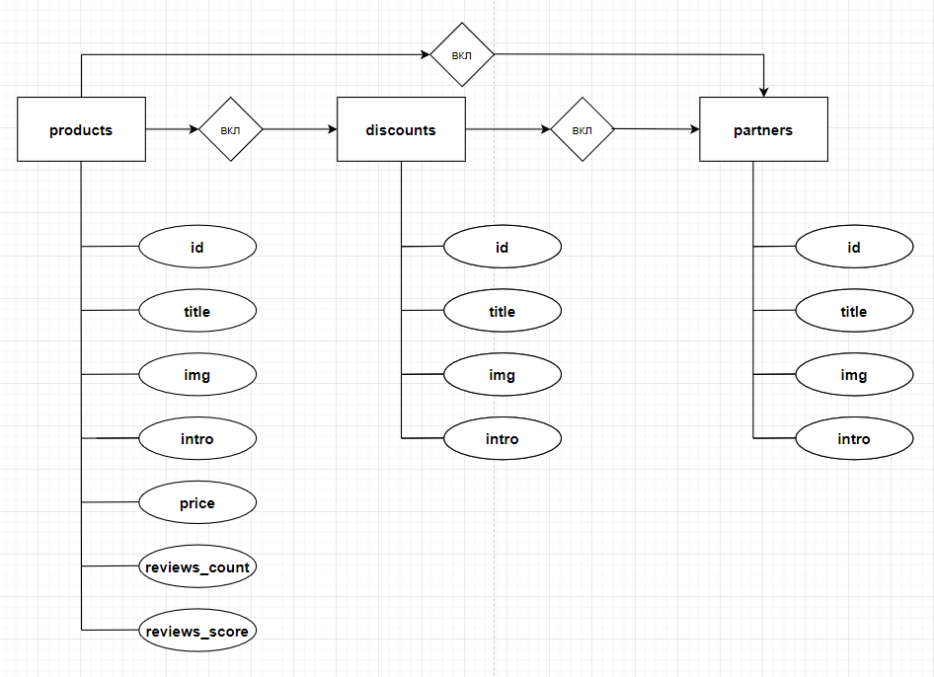


Рисунок 20- Концептуальная модель «Магазин игрушек Toy-store»

В разрабатываемой базе данных имеется информация о каждом конкрентном товаре(игрушке), что делает возможным обеспечение пользователя данной информацией. БД содержит в себе такие таблицы, как: «discounts», «partners», «products».Данные таблицы изображены на рисунках 22-24.

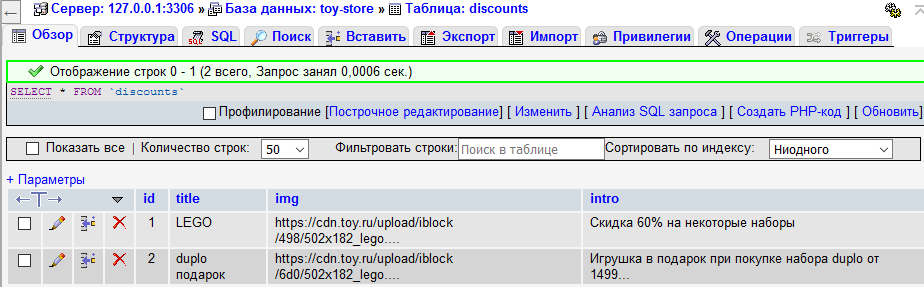


Рисунок 22-Таблица «discounts»

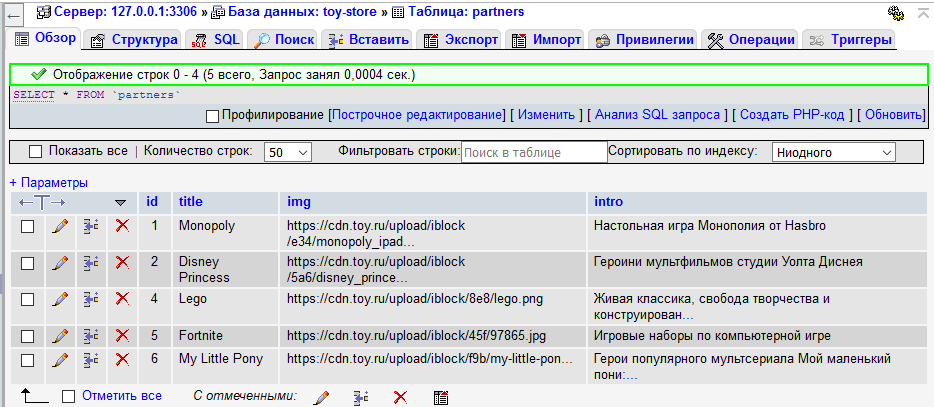


Рисунок 23-Таблица «partners»

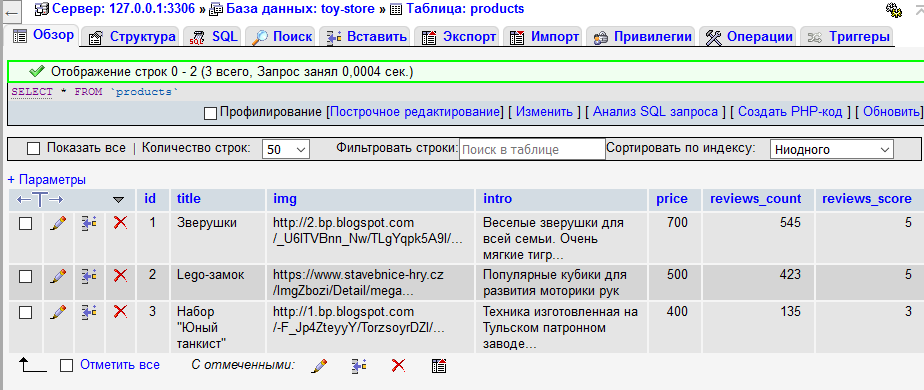


Рисунок 24-Таблица «product»

3.2 Разработка структуры БД

Разработка структуры БД предусматривает определение состава и взаимосвязи реляционных таблиц, описывающих предметную область, а также характеристику этих таблиц по входящим в них реквизитам.

Реквизит — единица информации, отражающая качественную или количественную характеристику объекта (процесса). Каждый реквизит базы данных имеет уникальное имя, а совокупность конкретных значений реквизита представляет в реляционной таблице столбец, который в терминах баз данных называется полем. Строки реляционной таблицы называются записями. Ключ — это реквизит или группа реквизитов, которые используются для идентификации конкретной записи в реляционной таблице. Реквизитный состав (анализ таблиц) представлен на рисунках 25-27:

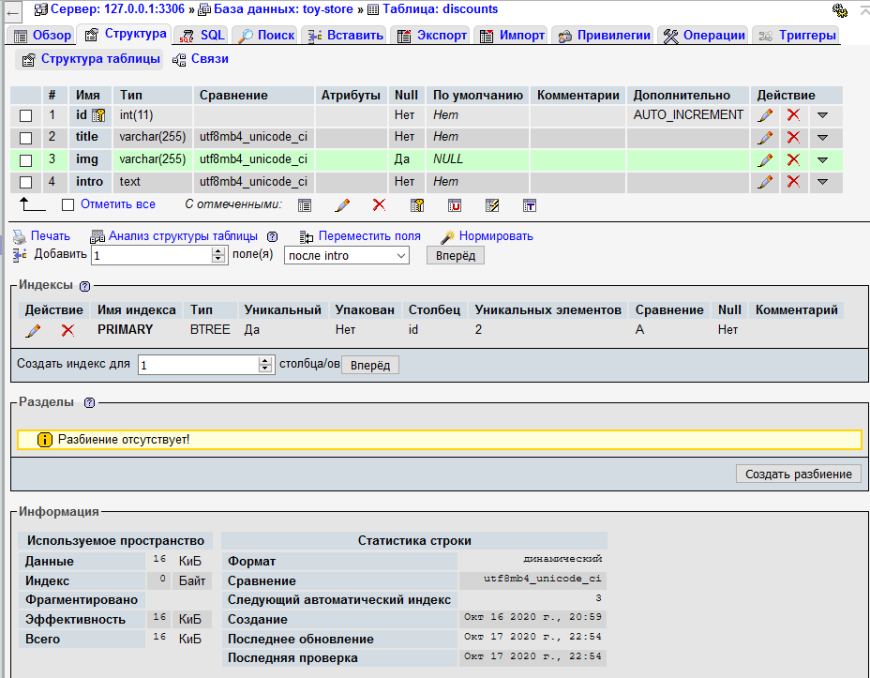


Рисунок 25 - Анализ таблицы «discounts»

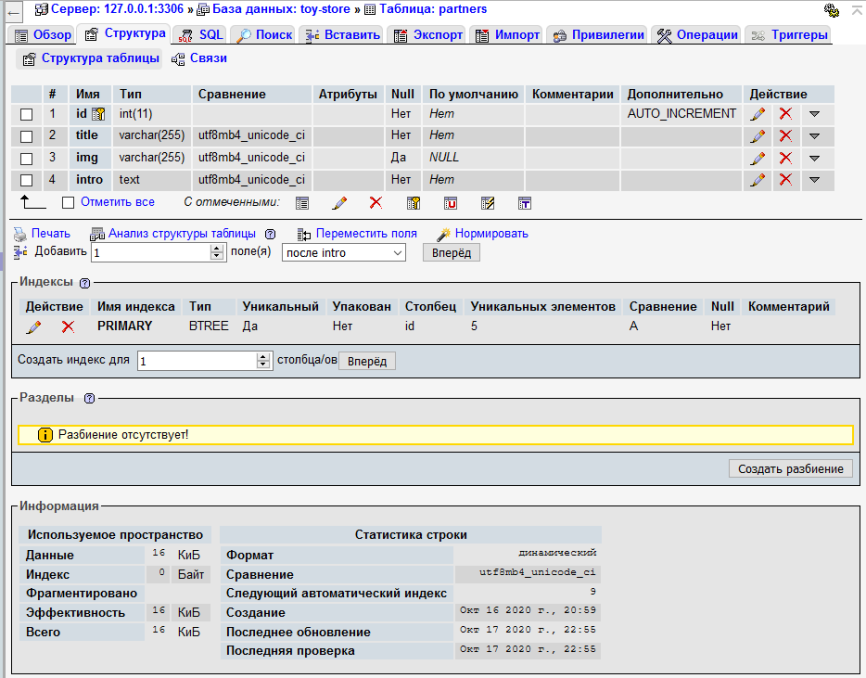


Рисунок 26 - Анализ таблицы «partners»

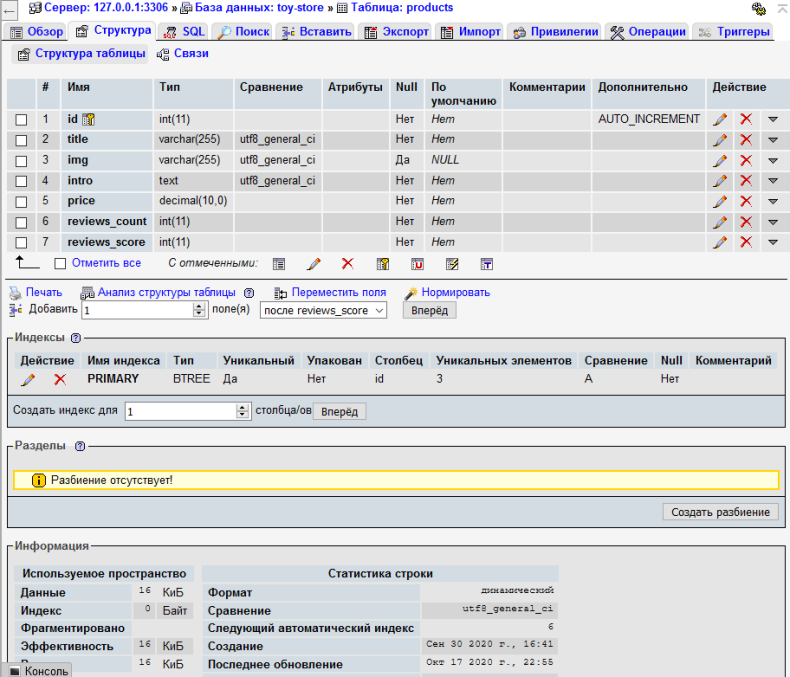


Рисунок 27 - Анализ таблицы «products»

## 3.3 Разработка системы управлением контентом

Исходя из анализа предметной области в системе необходимо разработать следующие интерфейсы:

1. Пользовательский интерфейс
2. Административный интерфейс

Администратор должен выполнять функции:

1. Добавлять записи в таблицы базы
2. Удалять и редактировать записи из таблиц

На рисунке 28 показан список пользователей с правами доступа к таблице



Рисунок 28- список пользователей с правами доступа к таблице

На рисунке 29 показан процесс создания пользователя с правами администратора

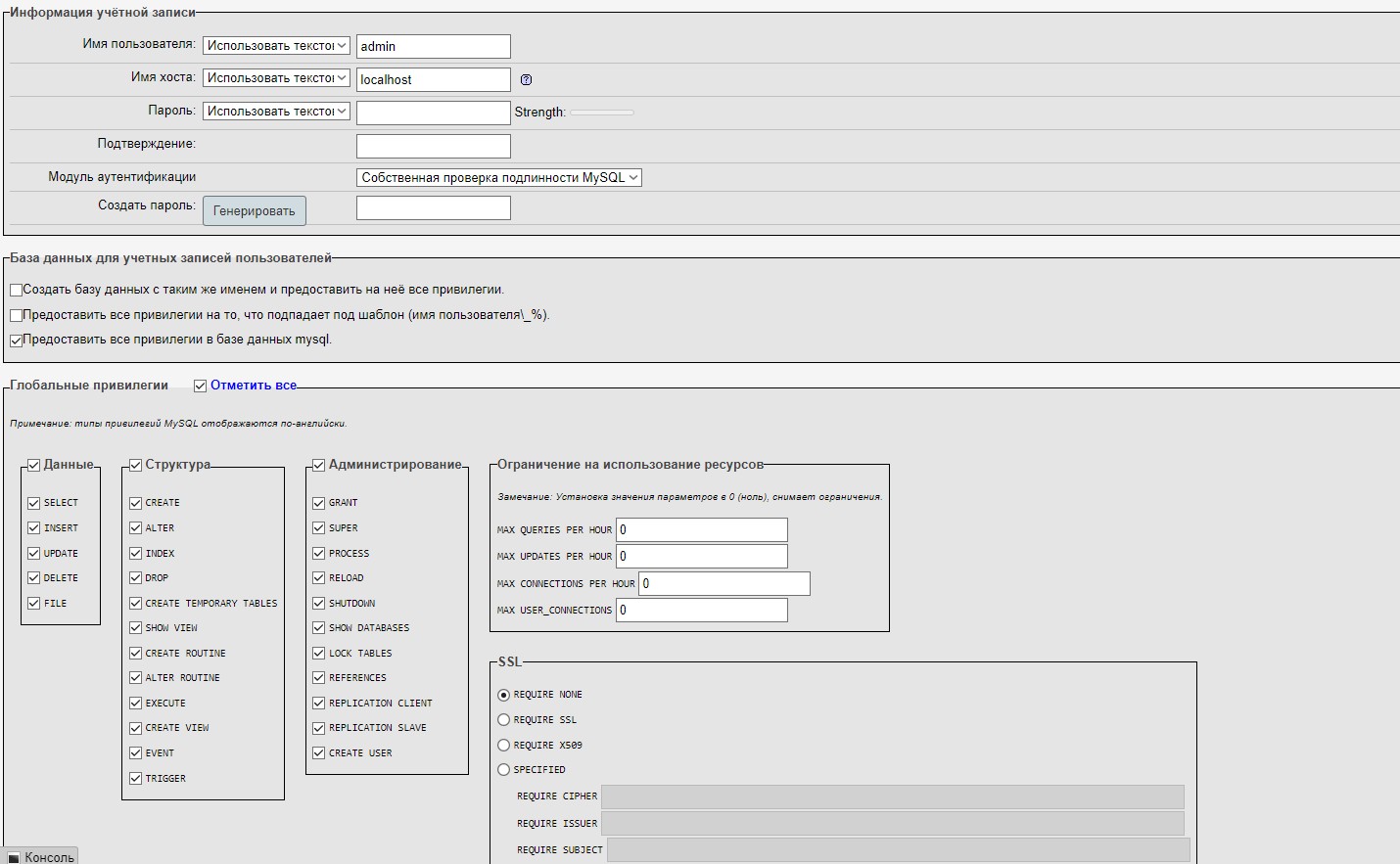


Рисунок 29- создания пользователя с правами администратора

Также мы можем редактировать права уже существующих пользователей. Процесс редактирования привелегий пользователя показан на рисунке 30

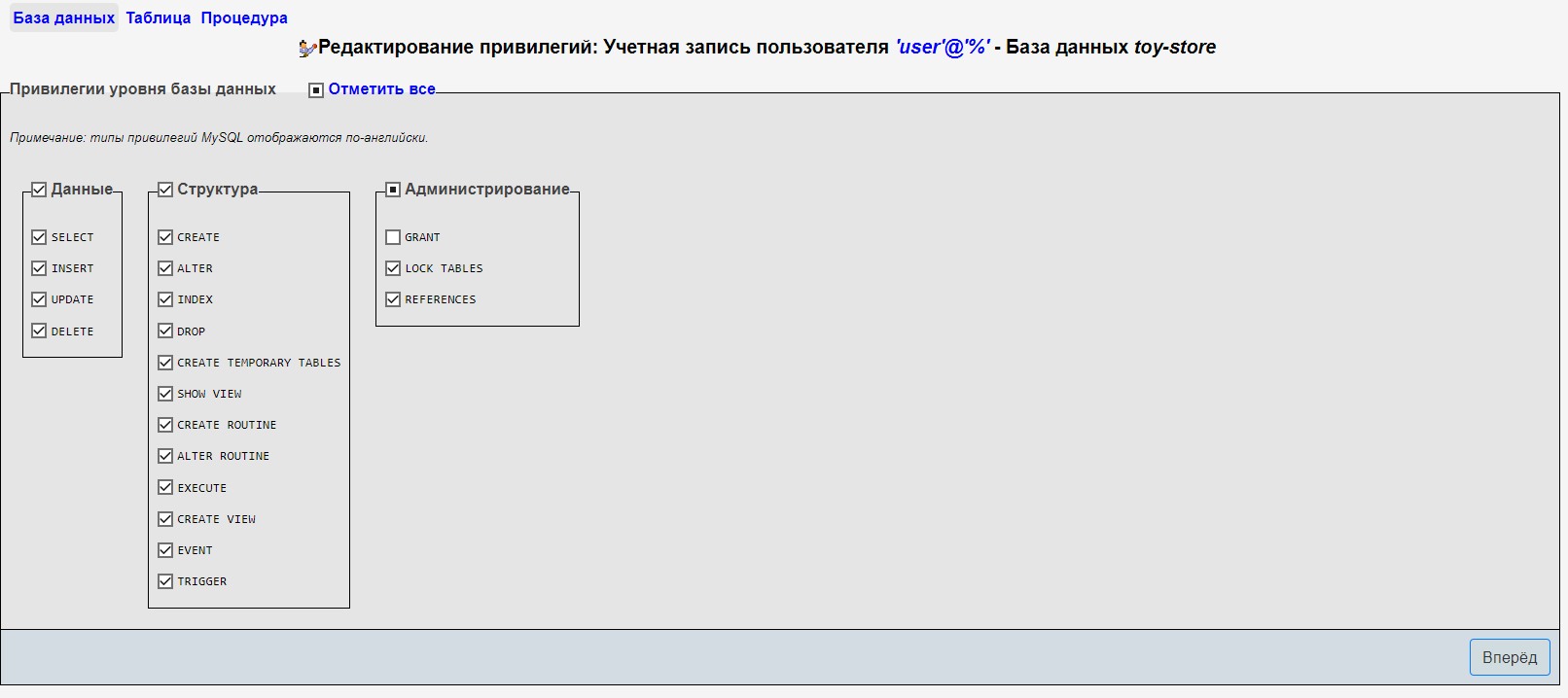


Рисунок 30 редактирование привилегей для пользователя

4 Система отслеживания ошибок

Также в работе был проделан анализ отслеживания найденных ошибок для исправления в программном обеспечении. На рисунках 31 представлена ошибка и неправильный пример связи.

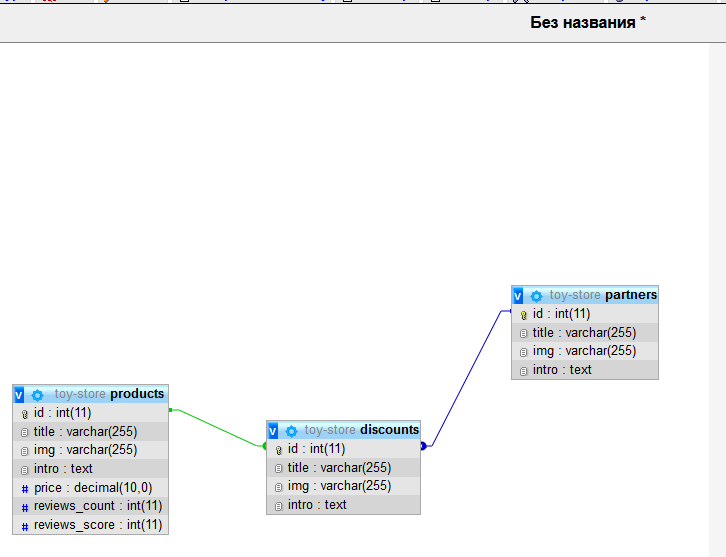


Рисунок 31-пример неправильной связи

Решение проблемы неправильной связи можно решить удалением связей(будет пустая схема связей) и созданием правильных связей. Решение представлено на рисунках 32-33

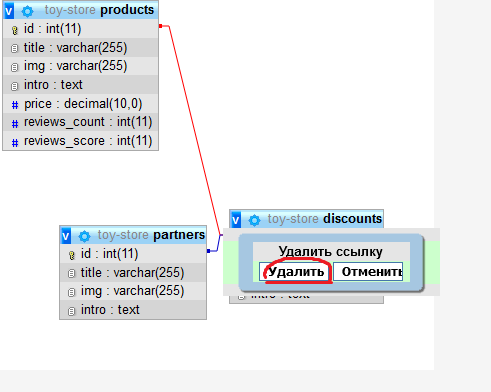


Рисунок 32-удаление связей

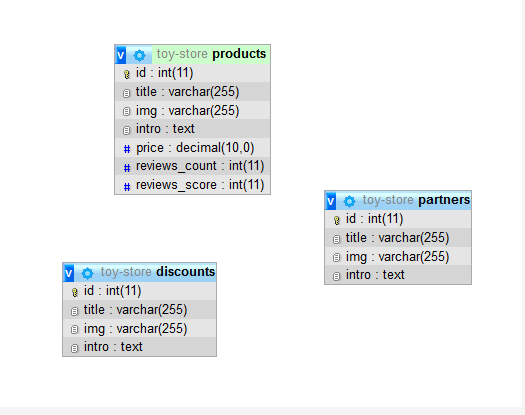


Рисунок 33-пустая схема после удаления связей

Ошибка при добавлении значений в таблицу discounts при не правильной работе связей.

Попытка ввода данных,ошибка представлены на рисунках 34-35

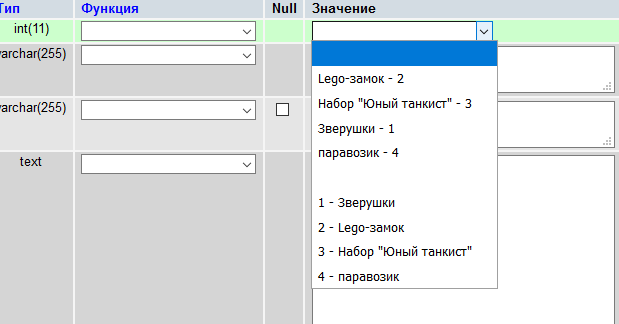


Рисунок 34-попытки добавления значений при не правильных связях

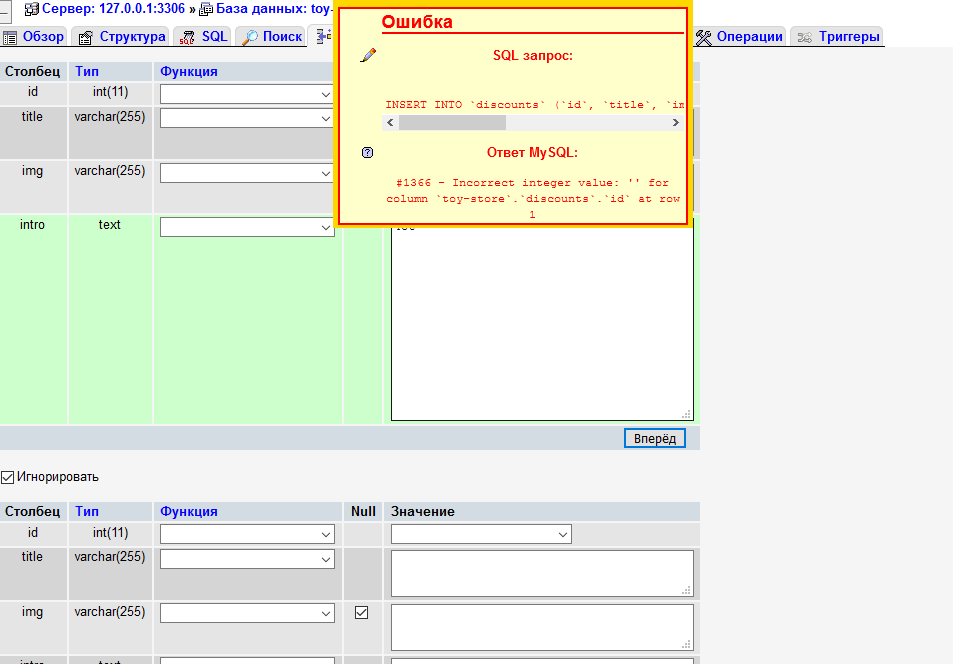


Рисунок 35-Ошибка при заполнении данных

Решение этой ошибки состоит в удалении и создании правильных связей которое было продемонстрировано на рисунках 32-33, затем ошибка исчезает и при вводе правильных значений мы видим результат.Попытка ввода правильных значений и результат после решения ошибки представлены на рисунках 36-37

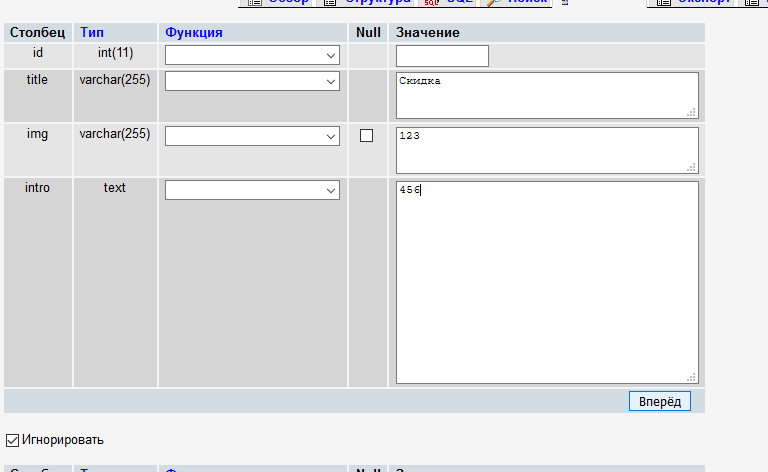


Рисунок 36-попытка ввода правильных значений

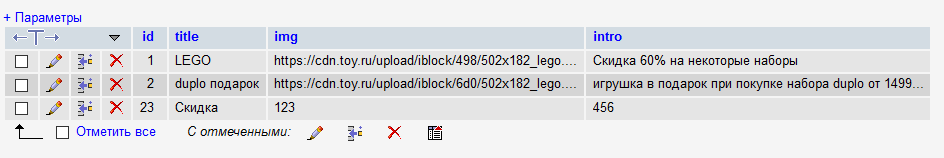


Рисунок 37-результат после решения проблемы

# ЗАКЛЮЧЕНИЕ

На практике ПМ.02 «Разработка и администрирование баз данных» были изучены и применены навыки создания серверной части базы данных. В частности изучена компактная и встраиваемая СУБД SQLite и софт для администрирования PhpMyAdmin. Полученные знания были применены для создания серверной части базы данных на архитектуре «клиент-сервер», её настройки и разработки системы управления контента веб-сайта toy-store.

Серверная часть базы данных является важным разделом компьютерных технологий поскольку выполняет облсуживание и управление баз данных и отвечает за целостность и сохранность данных, а также обеспечивает операции ввода-вывода при доступе клиента к информации.

Еще один важный раздел компьютерных технологий – это компьютерные технологии обучения, представляющие колоссальные возможности для повышения эффективности занятий, придания им интереса и динамичности.

По мере распространения компьютерной техники вся совокупность взаимодействующей с ней людей все более четко разделяются на две большие группы: системные и прикладные программисты.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Кузнецов Максим, Симдянов Игорь Объектно-ориентированное программирование на PHP[Письменный ресурс] -- Спб.: «БХВ-Петербург», 2007. - С. 608. –Книга
2. PhpMyAdmin. [Электронный ресурс]/ https://ipipe.ru/info/phpmyadmin - статья в интернете.
3. Дамашке Гизберт PHP и Sqlite [Письменный ресурс]; НТ Пресс - Москва, 2012. - 320 c.-книга
4. Остроух, А. В. Ввод и обработка цифровой информации[Письменный ресурс] / А.В. Остроух. - М.: Академия**, 2016**. - 288 c.
5. Мартишин, С. А. Базы данных. Практическое примечание СУБД SQL . Учебное пособие [Письменный ресурс]/ С.А. Мартишин, В.Л. Симонов, М.В. Храпченко. - М.: Форум, Инфра-М, 2016. - 368 c.