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](#_Toc53923984)

[1 GITHUB 8](#_Toc53923985)

[1.1 Распределённая система управления версиями. 8](#_Toc53923986)

[1.2 Преимущества Git 8](#_Toc53923987)

[1.2 Веб-сервис для хостинга и совместной разработки 9](#_Toc53923988)

[1.3 Сервер базы данных (SQL-сервер) 10](#_Toc53923989)

[1.4 Функционирование сервера базы данных 10](#_Toc53923990)

[2. СУБД MySQL 12](#_Toc53923991)

[2.1 Программа phpMyAdmin 12](#_Toc53923992)

[2.2 Определение состава данных 15](#_Toc53923993)

[2.3 Разработка концептуальной модели 16](#_Toc53923994)

[2.4 Разработка структуры БД 18](#_Toc53923995)

[ЗАКЛЮЧЕНИЕ 23](#_Toc53923996)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 24](#_Toc53923997)

# 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 MySQL является программным продуктом с открытым кодом. Обладает рядом преимуществ, таких как быстродействие, 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 MySQL.

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 MySQL архитектуры клиент – сервер.

# GITHUB

# Распределённая [система управления версиями](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D1%8F%D0%BC%D0%B8).

[Git](https://git-scm.com/about) — распределённая система контроля версий, которая даёт возможность разработчикам отслеживать изменения в файлах и работать совместно с другими разработчиками. Она была разработана в 2005 году Линусом Торвальдсом, создателем Linux, для того, чтобы другие разработчики могли вносить свой вклад в ядро Linux. Git известен своей скоростью, простым дизайном, поддержкой нелинейной разработки, полной децентрализацией и возможностью эффективно работать с большими проектами.

Системы контроля версий (СКВ, VCS, Version Control Systems) позволяют разработчикам сохранять все изменения, внесённые в код. Поэтому в случае, описанном выше, они могут просто откатить код до рабочего состояния вместо того, чтобы тратить часы на поиски маленькой ошибки или ошибок, ломающих весь код.

СКВ также дают возможность нескольким разработчикам работать над одним проектом и сохранять внесённые изменения, чтобы убедиться, что все могут следить за тем, над чем они работают.

Git стоит отдельно от других СКВ из-за подхода к работе с данными. Большинство других систем хранят информацию в виде списка изменений в файлах. Вместо этого, подход Git к хранению данных больше похож на набор снимков миниатюрной файловой системы. Каждый раз, когда вы сохраняете состояние своего проекта в Git, система запоминает, как выглядит каждый файл в этот момент, и сохраняет ссылку на этот снимок.

# 1.2 Преимущества Git

* **Бесплатный и open-source**. Это значит, что его можно бесплатно скачать и вносить любые изменения в исходный код;
* **Небольшой и быстрый**. Он выполняет все операции локально, что увеличивает его скорость. Кроме того, Git локально сохраняет весь репозиторий в небольшой файл без потери качества данных;
* **Резервное копирование**. Git эффективен в хранении бэкапов, поэтому известно мало случаев, когда кто-то терял данные при использовании Git;
* **Простое ветвление**. В других СКВ создание веток— утомительная и трудоёмкая задача, так как весь код копируется в новую ветку. В Git управление ветками реализовано гораздо проще и эффективнее.
* Основные команды.
* Всего несколько команд нужно для базового варианта использования Git для ведения истории изменений.
* git add. Команда git add добавляет содержимое рабочей директории в индекс (staging area) для последующего коммита. По умолчанию git commit использует лишь этот индекс, так что вы можете использовать git add для сборки слепка вашего следующего коммита.
* git status. Команда git status показывает состояния файлов в рабочей директории и индексе: какие файлы изменены, но не добавлены в индекс; какие ожидают коммита в индексе. Вдобавок к этому выводятся подсказки о том, как изменить состояние файлов.
* git diff. Команда git diff используется для вычисления разницы между любыми двумя Git деревьями.
* git difftool. Команда git difftool просто запускает внешнюю утилиту сравнения для показа различий в двух деревьях, на случай если вы хотите использовать что-либо отличное от встроенного просмотрщика git diff.
* git commit. Команда git commit берёт все данные, добавленные в индекс с помощью git add, и сохраняет их слепок во внутренней базе данных, а затем сдвигает указатель текущей ветки на этот слепок.
* git clean. Команда git clean используется для удаления мусора из рабочей директории. Это могут быть результаты сборки проекта или файлы конфликтов слияний.

# 1.2 [Веб-сервис](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B8%D1%81) для [хостинга](https://ru.wikipedia.org/wiki/%D0%A5%D0%BE%D1%81%D1%82%D0%B8%D0%BD%D0%B3) и совместной разработки

[GitHub](https://github.com/) — сервис онлайн-хостинга репозиториев, обладающий всеми функциями распределённого контроля версий и функциональностью управления исходным кодом — всё, что поддерживает Git и даже больше. Обычно он используется вместе с Git и даёт разработчикам возможность сохранять их код онлайн, а затем взаимодействовать с другими разработчиками в разных проектах.

Также GitHub может похвастаться контролем доступа, багтрекингом, управлением задачами и вики для каждого проекта. Цель GitHub — содействовать взаимодействию разработчиков.

К проекту, загруженному на GitHub, можно получить доступ с помощью интерфейса командной строки Git и Git-команд. Также есть и другие функции, такие как документация, запросы на принятие изменений (pull requests), история коммитов, интеграция со множеством популярных сервисов, email-уведомления, эмодзи, графики, вложенные списки задач, система @упоминаний.

Git — это инструмент, позволяющий реализовать распределённую систему контроля версий, а GitHub — это сервис для проектов, использующих Git.

# 1.3 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.

# 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.

2 СУБД MYSQL

СУБД MySQL является программным продуктом с открытым кодом. Обладает рядом преимуществ, таких как быстродействие, соответствие стандартным требованиям, совместимостью с аналогичными системами. В качестве основного инструмента разработки баз данных 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 MySQL. Приложения и интерфейс базы данных выполнен в скриптах языка PHP. Проект выполнен в соответствии с рабочей программой ПМ.02«Разработка и администрирование баз данных». Тема данной работы (Магазин игрушек). В работе представлены основные этапы анализа и разработки предметной области (субд), разработка концептуальной модели базы и физической модели базы. В качестве СУБД выбрана система MySQL архитектуры клиент – сервер.

2.1 Программа phpMyAdmin

Программа phpMyAdmin является одним из самых распространенных средств для администрирования, управления и каждодневной работы с базами данных MySQL. 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 MySQL. В дополнение к возможностям самого сервера MySQL, 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.

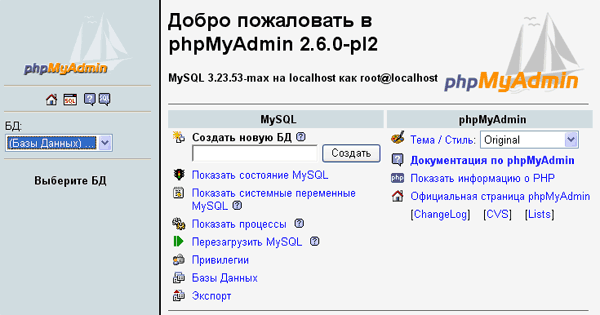


Рисунок 1 –Интерфейс phpMyAdmin

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. Конечно, до отладчика из пакета MySQL Query Browser phpMyAdmin еще очень далеко, но это ведь совершенно различные весовые категории.

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 2.

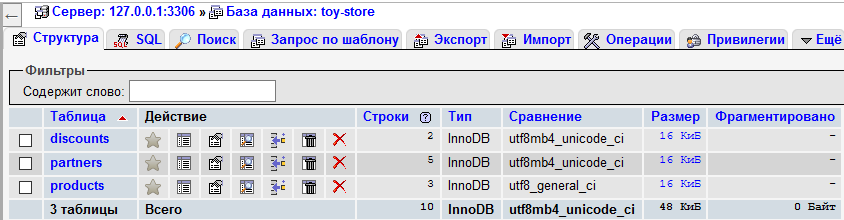


Рисунок 2 - Структура таблиц базы данных Toy-store.

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.

2.2 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 СУБД.

На рисунке 3 представлена физическая модель базы данных со связями между таблицами

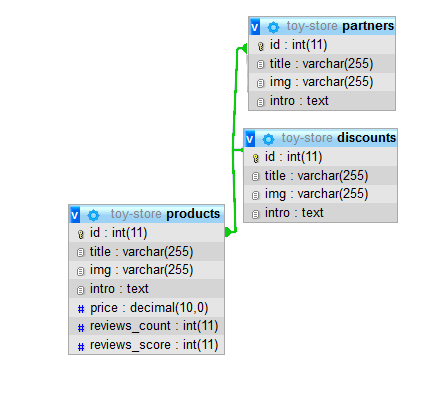


Рисунок 3 - Связи таблиц базы данных «Магазин игрушек Toy-store»

2.3 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 4 представлена концептуальная модель базы данных.

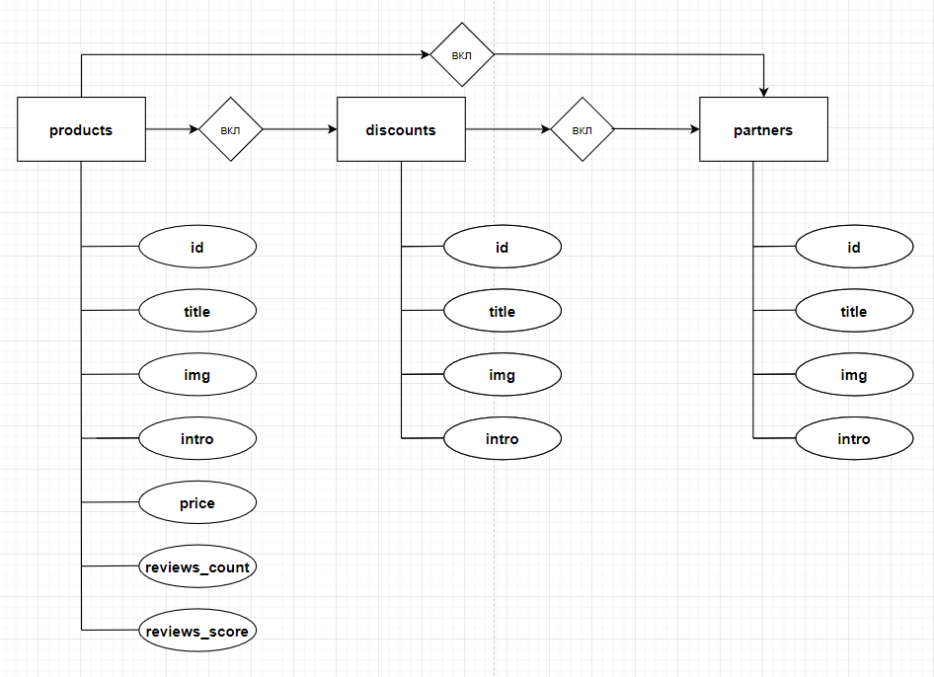


Рисунок 4- Концептуальная модель «Магазин игрушек Toy-store»

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 5-7.

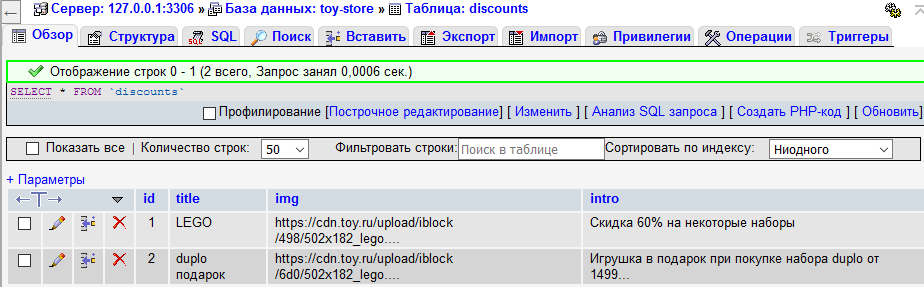


Рисунок 5-Таблица «discounts»

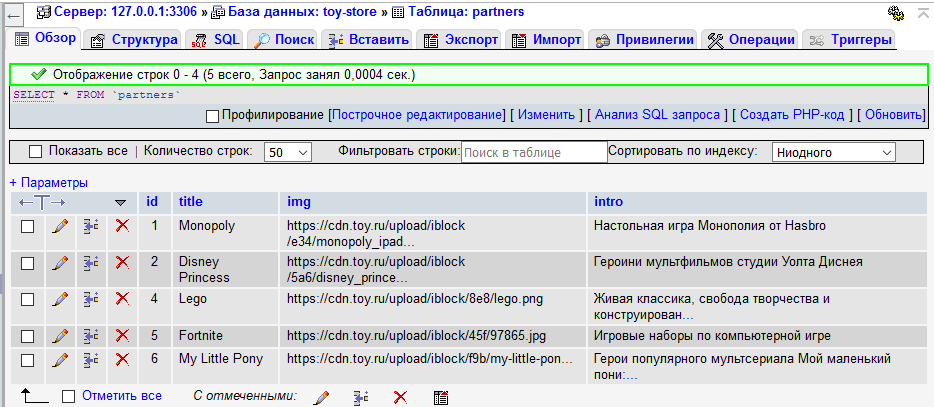


Рисунок 6-Таблица «partners»

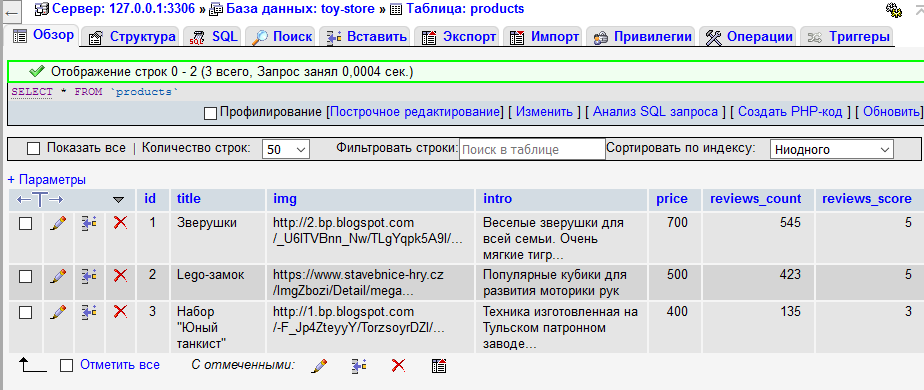


Рисунок 7-Таблица «product»

2.4 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 8-10:

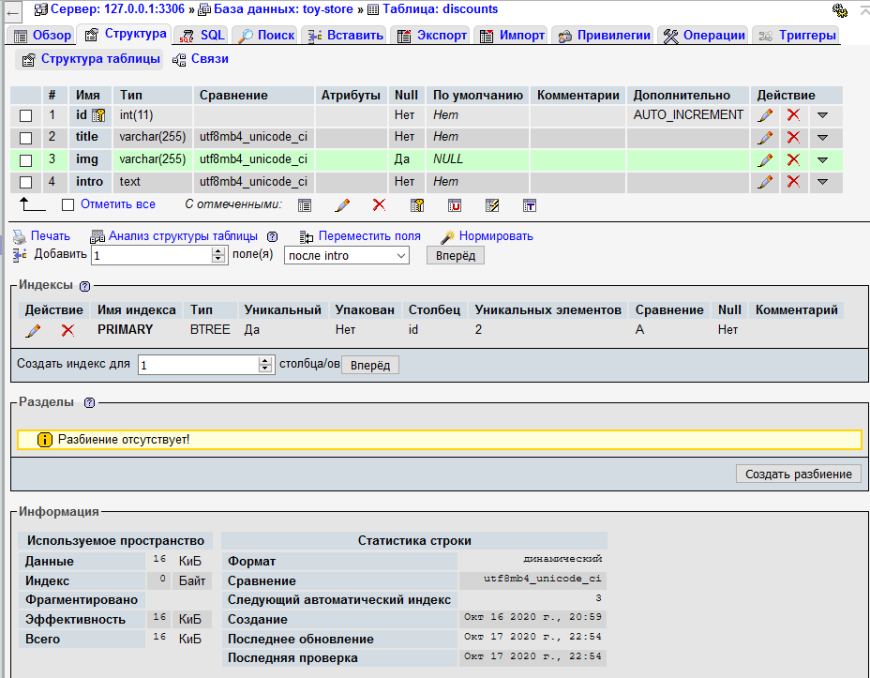


Рисунок 8 - Анализ таблицы «discounts»

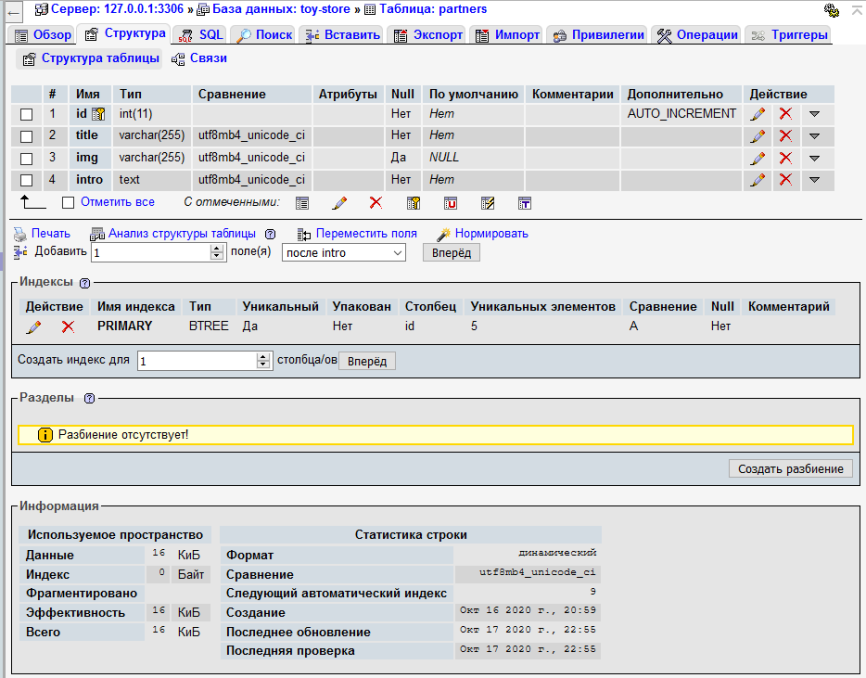


Рисунок 9 - Анализ таблицы «partners»

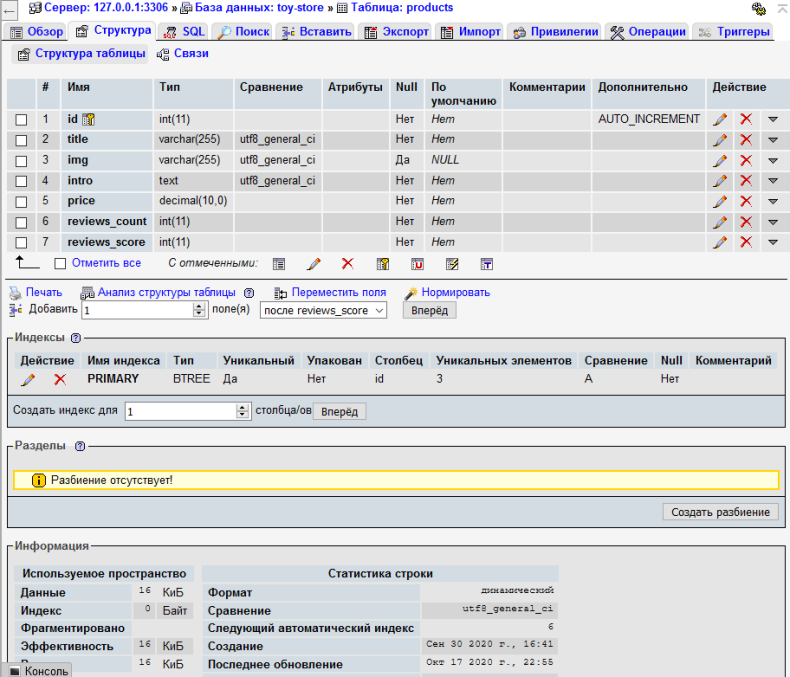


Рисунок 10 - Анализ таблицы «products»

2.5 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 11 показан процесс создания учётной записи администратора

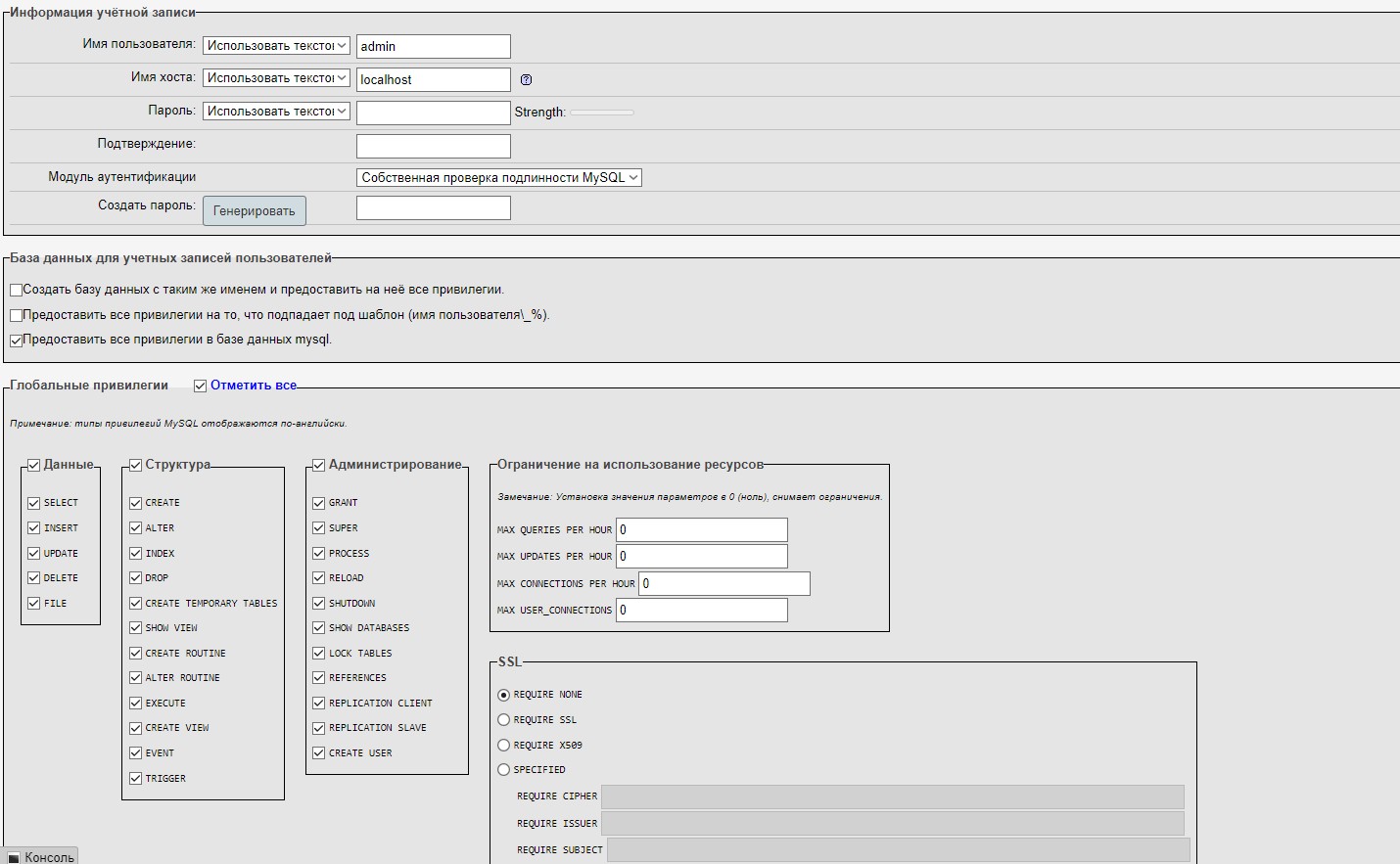


Рисунок 11- Создание пользователя с привилегиями

# ЗАКЛЮЧЕНИЕ

Мы рассмотрели большинство основных способов настройки клиента и сервера Git с тем, чтобы он был максимально соответствовал вашим рабочим процессам и проектам.

Мы узнали о всевозможных настройках, атрибутах файлов и о перехватчиках событий, а также рассмотрели пример настройки сервера с соблюдением политики. Теперь вам должно быть по плечу заставить Git подстроиться под практически любой тип рабочего процесса, который только можно вообразить.

Повсеместного распространения электронных вычислительных машин (ЭВМ) человеческие знания о природе информации приобретают общекультурную ценность. Этим объясняется интерес исследователей и практиков всего мира к относительно молодой и быстро развивающейся научной дисциплине - информатике. На сегодняшний день информатика выделилась в фундаментальную науку, и она не может быть сведена к другим наукам. Объектом изучения информатики являются структура информации и методы ее обработки.

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. .Git и Githab: что это такое и чем отличается [Электронный ресурс]/ https://tproger.ru/translations/difference-between-git-and-github/- статья в интернете.
2. Кузнецов Максим, Симдянов Игорь Объектно-ориентированное программирование на PHP[Письменный ресурс] -- Спб.: «БХВ-Петербург», 2007. - С. 608. –Книга
3. PhpMyAdmin. [Электронный ресурс]/ https://ipipe.ru/info/phpmyadmin - статья в интернете.
4. Дамашке Гизберт PHP и MySQL [Письменный ресурс]; НТ Пресс - Москва, 2012. - 320 c.-книга