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\_\_\_\_\_\_\_\_\_Чернухин А.М.\_\_ / ( \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ )**

**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 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Чернухин Алексей Михайлович\_\_\_\_\_\_\_\_\_\_\_

*ФИО*

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

успешно прошел(ла) 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](#_Toc54115632)

[1 PHPMyAdmin 8](#_Toc54115633)

[2 СОЗДАНИЕ И УДАЛЕНИЕ БАЗ ДАННЫХ 11](#_Toc54115634)

[3. ТЕСТИРОВАНИЕ ЗАПРОСОВ 17](#_Toc54115635)

[4. РЕДАКТИРОВАНИЕ ЗАПИСЕЙ В БД 19](#_Toc54115636)

[5 СИСТЕМА ОТСЛЕЖИВАНИЯ ОШИБОК 20](#_Toc54115637)

[ЗАКЛЮЧЕНИЕ 23](#_Toc54115638)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 24](#_Toc54115639)

[ПРИЛОЖЕНИЯ 25](#_Toc54115640)

# ВВЕДЕНИЕ

Практическая выполнена в соответствии с рабочей программой ПМ.02 «Разработка и администрирование баз данных». В работе представлены основные этапы анализа и разработки предметной области «Разработка и администрирование базы данных», навыки работы с phpMyAdmin.

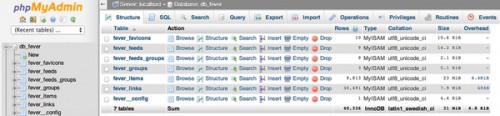
База данных играет важную роль в большинстве предметных областей. Благодаря динамической природе автоматизированных отраслей сейчас приложения требуют некоторых механизмов хранения, доступа и изменения данных. Поскольку важность баз данных стремительно растёт, реляционные системы управления базами данных набирают свою популярность.

В качестве программного обеспечения для создания интерфейса выбран редактор Notepad++. А для написания кода выбраны: язык гипертекстовой разметки HTML, язык таблиц стилей CSS и для работы с базами данных PHP. Использован веб-интерфейс phpmyadmin.

В процессе выполнения проекта должны быть получены знания и опыт в области в технологии phpMyAdmin. Также мною будут решаться вопросы администрирования базы данных и реализовываться методы и технологии её защиты.

# 1 [PHPMyAdmin](http://www.phpmyadmin.net/home_page/index.php)

[PHPMyAdmin](http://www.phpmyadmin.net/home_page/index.php) (**PMA**) — превосходный бесплатный веб-клиент баз данных с открытым исходным кодом, который может использоваться для более активного взаимодействия с **MySQL**и приложениями баз данных.



Я расскажу, как его установить, обеспечить безопасность, а также опишу некоторые общие сценарии, при которых он может помочь в администрировании баз данных. Здесь для вашего ознакомления размещено [онлайн демо PMA](http://demo.phpmyadmin.net/STABLE/?pma_username=root).

В дополнение к визуальному графическому интерфейсу для операций с базами данных, я также высоко ценю возможность запуска операций из командной строки **SQL** моего браузера. Без необходимости непосредственно заходить на сервер через **SSH**.

Например, некоторые **Wi-Fi** соединения и соединения мобильных точек доступа регулярно обрывают сеансы **SSH**, что делает работу с базой данных весьма проблематичной.

Установка PHPMyAdmin

Заходим на сервер через **SSH**:

Настройте веб-аутентификацию для сайта **PMA**.

Для этого потребуется ввести дополнительный пароль, чтобы получить доступ к **PMA**, в дополнение к вашему паролю к базе данных, например:

apt-get install phpmyadmin

Во время установки вы можете использовать настройки по умолчанию или настроить клиент по своему вкусу.

После того как вы ограничите доступ к MySQL только с одного локального хоста (своего), в MySQL больше не останется удаленных портов, которыми бы могли воспользоваться хакеры.

Они, конечно, могут попытаться прорваться через SSH или попробовать против ваших приложений атаки внедрения SQL, но непосредственно базу данных они атаковать не смогут. После установки PMA, никто не сможет осуществить веб-атаку, чтобы получить контроль над вашей базой данных, так что безопасность будет обеспечена.

Существует несколько мер предосторожности, которые я рекомендую при настройке PMA:

• Используйте очень надежные пароли для всех ваших учетных записей MySQL, особенно для аккаунта root. Например, 25 символов для пароля root;

• Используйте различные учетные записи и привилегии MySQL для каждого приложения / сайта, работающего на одном сервере. Таким образом, если один пароль будет взломан, под угрозой окажется база данных только одного сайта;

• Измените URL-адрес, используемый PMA по умолчанию. Таким образом, люди не смогут посещать http://адрес\_вашего\_сайта /phpmyadmin. Хотя это и не является сверх эффективным методом, он все равно несколько повышает безопасность.

Добавьте псевдонимы в файл apache.conf:

nano /etc/phpmyadmin/apache.conf

Alias /myobscuredpma /usr/share/phpmyadmin

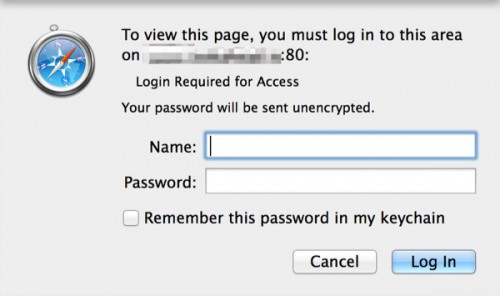
Перегрузите Apache:

service apache2 reload

Затем, чтобы получить доступ к PMA, зайдите на http://адрес\_вашего\_сайта/myobscuredpma.

Если вам нужно изменить ваш пароль PHPMyAdmin, вы можете отредактировать файл config-db.php здесь:

nano /etc/phpmyadmin/config-db.php



Чтобы настроить ограничения пользователей Apache, выполните следующие действия:

Установите htpasswd как часть apache2-utils:

apt-get install apache2-utils

Создайте папку для хранения паролей:

mkdir /etc/htpasswd

Лучшие редакторы кода и IDE для разработки на PHP в 2017 году

Добавьте поддержку htaccess для PMA:

nano /etc/phpmyadmin/apache.conf

Добавьте AllowOverride All:

<Directory /usr/share/phpmyadmin>

Options FollowSymLinks

DirectoryIndex index.php

AllowOverride All

Настройте аутентификацию пользователей для PMA:

nano /usr/share/phpmyadmin/.htaccess

AuthName "Login Required for Access"

Require valid-user

Установите ваш пароль:

htpasswd -c /etc/htpasswd/.htpasswd username

И перезапустите Apache:

service apache2 restart

Зайдите через браузер на ваш PMA-сайт, и вам будет предложено ввести имя пользователя и пароль, как показано выше.

# 2 СОЗДАНИЕ И УДАЛЕНИЕ БАЗ ДАННЫХ

Я считаю, PMA особенно полезен на стадии разработки и тестирования, где, возможно, вам неоднократно потребуется сбросить базу данных или восстановить ее из резервной копии.

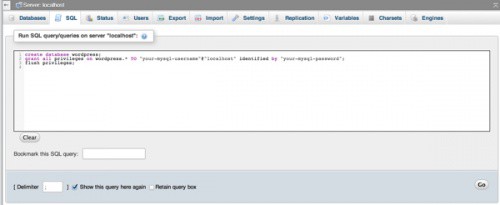
Без PMA, мне нужно было бы заходить на свой сервер через SSH. Затем заходить в MySQL, а затем запускать:

create database myapp;

grant all privileges on myapp\_database.\* TO "your-mysql-username"@"localhost" identified by "your-mysql-password";

flush privileges;

С PMA вы можете запустить любой запрос командной строки через визуальный интерфейс. Перейдите на вкладку **SQL** и вставьте инструкции для базы данных **MySQL**. После этого нажмите кнопку **Go** для создания базы данных:

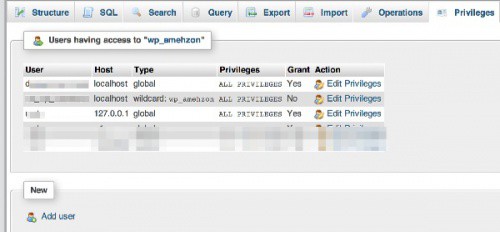


Установка пароля на страницу

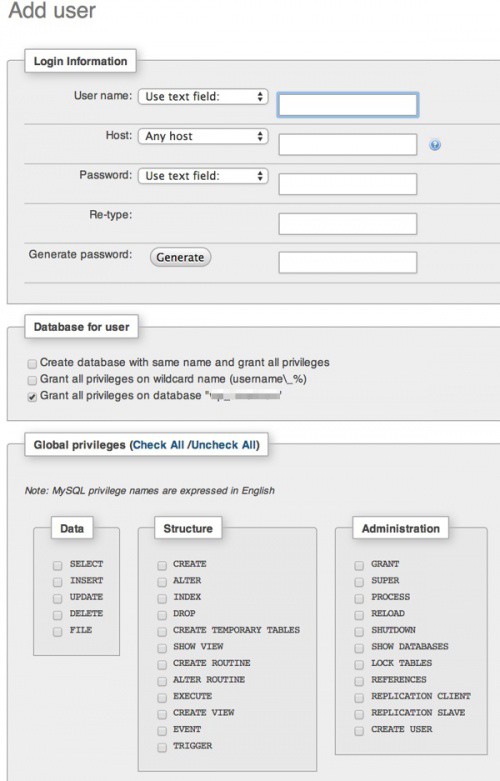
Или, вы можете использовать визуальный интерфейс напрямую:



Вы также можете добавить пользователей и определить для них привилегии. Перейдите на вкладку привилегий:

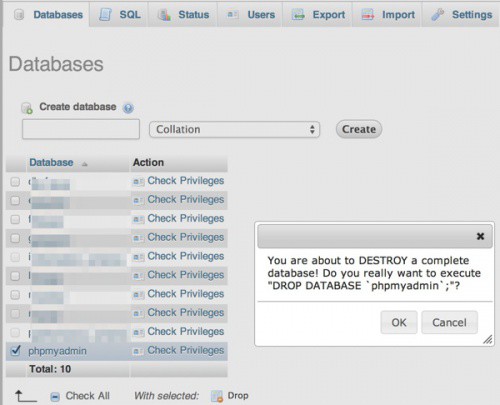


Нажмите кнопку добавления пользователя и определите ему привилегии для нужной базы данных:



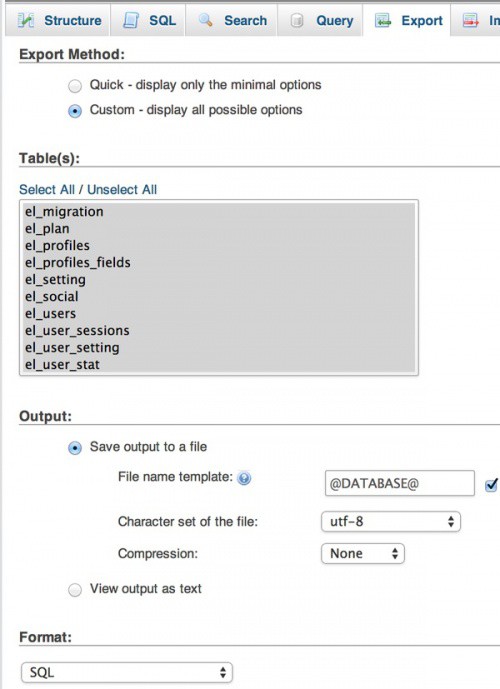
Для вашего основного аккаунта приложений баз данных, вам нужно будет проверять панели разделов «Данные» и «Структура».

Чтобы удалить базу данных, воспользуйтесь меню «Базы данных». Выберите базу данных, которую вы хотите удалить, и нажмите «Удалить»:

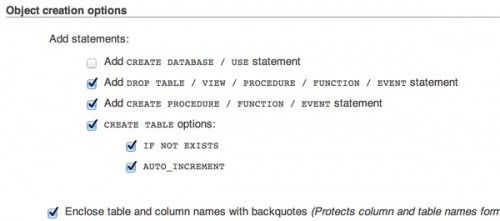


**Резервное копирование базы данных -** Перед выполнением любых операций по расширению базы данных или обновлению кода, всегда следует сохранять базу данных приложения. Если что-то пойдет не так, вы всегда сможете восстановить свой сайт из резервной копии.

С помощью веб-интерфейса **PMA**, выберите базу данных, выберите вкладку **Export** и выберите пункт **Custom**:



Установка защиты на страницу используя MySQL и PHP  
Включите оператор **«Add Drop Table / View / Procedure / Function / Event»**:

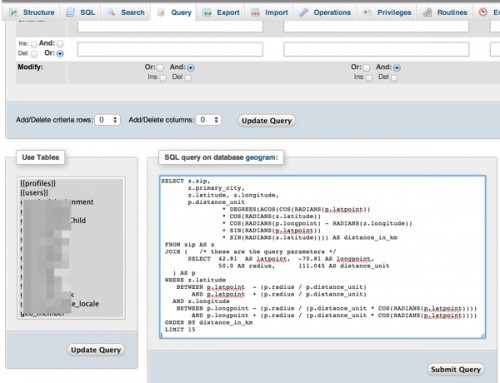


При нажатии **Go PMA** загрузит резервную копию всей вашей базы данных. Если параметры задержки **Apache PHP** настроены неправильно, загрузка некоторых больших файлов может зависнуть и оборваться. Вы можете настроить это в **PHP.ini**, [как я описал на своем сайте](http://publishingwithwordpress.com/moving-wordpress/#Use_Duplicator_to_Backup_Your_Site).

# **3. ТЕСТИРОВАНИЕ ЗАПРОСОВ**

**PMA**очень здорово помогает во время разработки при тестировании запросов **SQL**. Когда я работал над [Geogram](http://geogram.com/" \t "_blank) мне нужно было изучить и проверить ряд сложных запросов геолокации по расположенным поблизости объектам, например, найти объекты поблизости от моего адреса. С **PMA** это было сделать намного проще.

Выберите базу данных, нажмите **Query**. Вставьте или отредактируйте сложные запросы и проверьте их непосредственно из **PMA**:



После того, как я установил, что мои запросы работают, я очень легко смог написать программный код **ActiveRecord**. В качестве примера:

$criteria = new CDbCriteria;

$criteria->together=true;

$criteria->having= "distance < 100";

$criteria->order = "distance ASC";

$criteria->with = array('place\_geometry'=>array('select'=>'place\_id,center,'.new

CDbExpression('(3959 \* acos( cos( radians('.$lat.') ) \* cos( radians( X(center) ) ) \* cos( radians( Y(center) ) - radians('.$lon.') ) + sin( radians('.$lat.') ) \* sin( radians( X(center) ) ) )) as distance')));

$dataProvider = new CActiveDataProvider(Place::model()->active()->includesMember($id),

array('criteria' => $criteria,

'pagination' => array(

'pageSize' => 10,

),

));

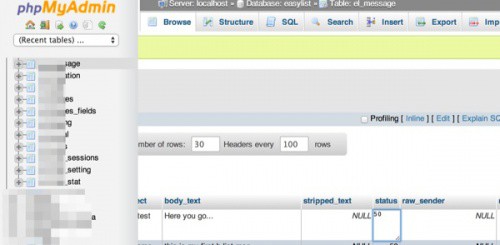
# **4. РЕДАКТИРОВАНИЕ ЗАПИСЕЙ В БД**

Если вы разработчик, вы, вероятно, сталкивались с ситуацией, когда в результате ошибки приложения ваша база данных была повреждена.

Например, некоторые из моих приложений управления электронной почтой иногда дают сбой при обработке необычных сообщений; эти ошибки трудно обнаружить при обычном тестировании.

Когда это произошло, я счел полезным изменить базу данных непосредственно через **PMA**, отсоединить задачи и снова запустить сайт. В некоторых случаях возможность просто просматривать таблицы может помочь вам при диагностике неполадок.

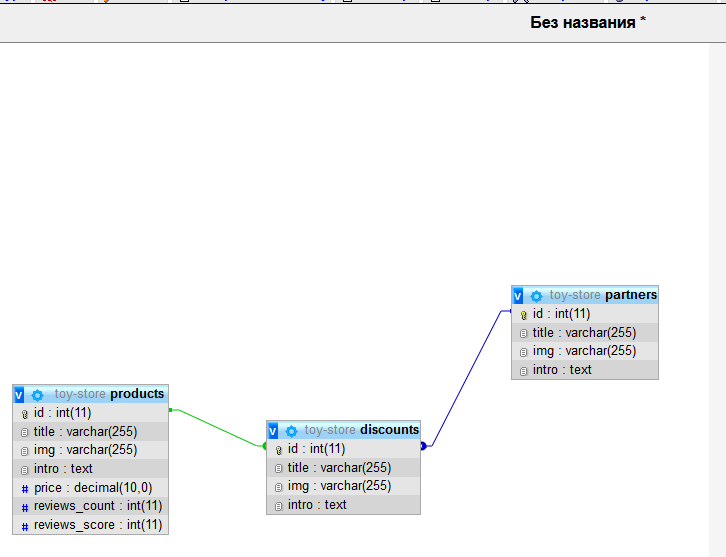
В **PMA** откройте базу данных и выберите таблицу. Дважды щелкните на столбце, который вы хотите отредактировать. Внесите нужные изменения и вернитесь обратно:



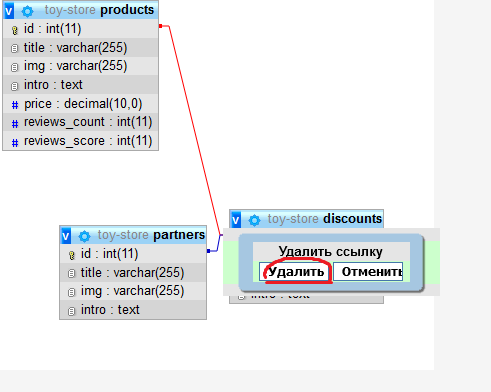
# 

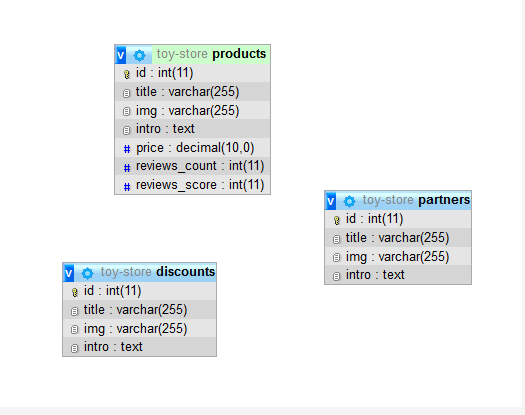
# 5 СИСТЕМА ОТСЛЕЖИВАНИЯ ОШИБОК

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



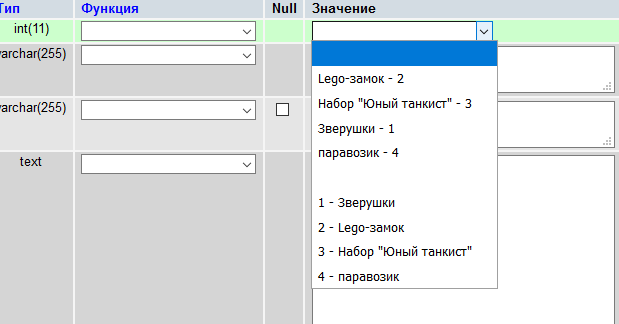
Решение проблемы неправильной связи можно решить удалением связей (будет пустая схема связей) и созданием правильных связей. Решение представлено наследующих рисунках.

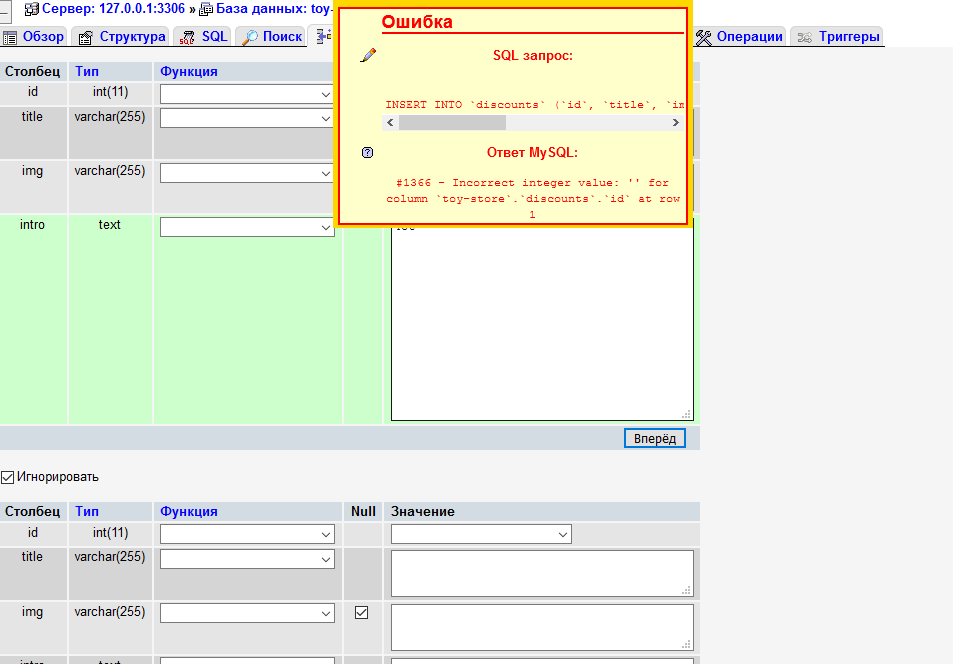




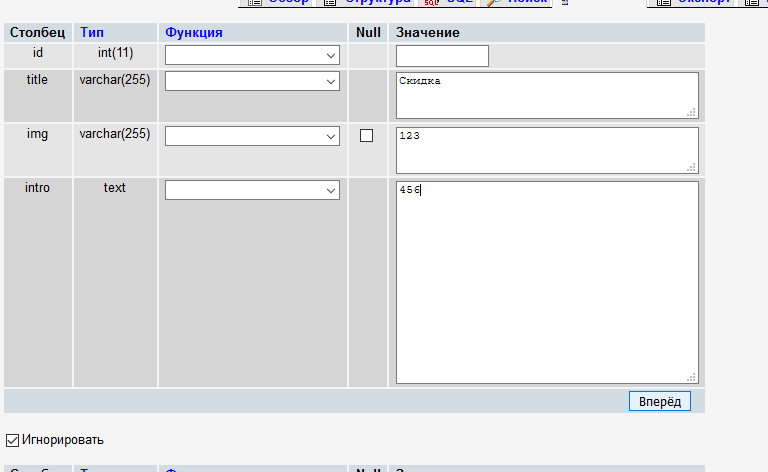
Ошибка при добавлении значений в таблицу discounts при не правильной работе связей.

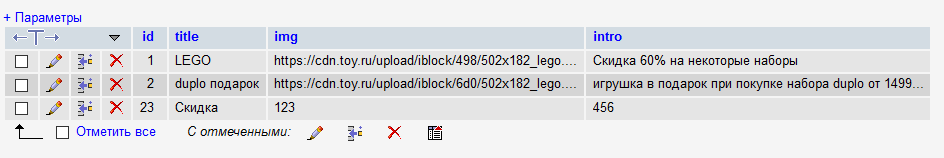
Попытка ввода данных, ошибка представлены на следующих рисунках





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





# ЗАКЛЮЧЕНИЕ

Я рассмотрел основные моменты по установке и настройке phpmyadmin.

В заключение следует отметить, что phpMyAdmin не самое простое приложение для управления БД. Огромный функционал является плюсом и в то же время делает GUI довольно громоздким. Например, не сразу понятно, когда рабочая область настроена для работы с конкретной БД (если она выбрана), а когда глобально. Из полезных особенностей стоит выделить сопровождение всех действий всплывающими сообщениями об успешности или ошибках. В которых приводится текст SQL-запроса, соответствующий этим действиям. Таким образом можно на практике осваивать язык SQL.

В процессе выполнения проекта были получены знания и опыт в области в технологии phpMyAdmin. Также мною были решены вопросы администрирования базы данных и реализованы методы и технологии её защиты.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Github: [Электронный ресурс]/https://tproger.ru/translations/difference-between-git-and-github - статья в интернете.
2. Инструкция по установке php: [Электронный ресурс]/ https://www.internet-technologies.ru/ - статья в интернете.
3. phpMyAdmin: [Электронный ресурс] /https://www.phpmyadmin.net/ - статья в интернете.

# ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А

<?php

session\_start();

if($\_GET['do'] == 'logout'){

unset($\_SESSION['admin']);

session\_destroy();

}

if($\_SESSION['admin'] != "admin"){

header("Location: login.php");

exit;

}

?>

<?php

$mysqli = mysqli\_init();

if (!$mysqli) {

die('mysqli\_init завершилась провалом');

}

if (!$mysqli->options(MYSQLI\_OPT\_CONNECT\_TIMEOUT, 5)) {

die('Установка MYSQLI\_OPT\_CONNECT\_TIMEOUT завершилась провалом');

}

?>

<?php

$products = array();

if ($result = $mysqli->query {

while($tmp = $result-> {

$products[] = $tmp;

}

$result->close();

}

?>

<?php

session\_start();

$users = 'admin';

$pass = 'rktmai';

if($\_POST['submit']){

if($users == $\_POST['user'] AND $pass == ($\_POST['pass']))

{

$\_SESSION['admin'] = $users;

header("Location: index.php");

exit;

}

$users2 = '123';

pass2 = '123';

if($\_POST['submit']){

if($users2 == $\_POST['user'] AND $pass2 == ($\_POST['pass']))

{

$\_SESSION['123'] = $users;

header("Location: admin.php");

exit;

}

}

else echo '<p>Логин или пароль неверны!</p>';

}

?>

<?php

$mysqli = mysqli\_init();

if (!$mysqli) {

die('mysqli\_init завершилась провалом');

}

if (!$mysqli->options(MYSQLI\_INIT\_COMMAND, 'SET AUTOCOMMIT = 0')) {

die('Установка MYSQLI\_INIT\_COMMAND завершилась провалом');

}

?>

<?php

$products = array();

if ($result = $mysqli->query('SELECT \* FROM partners')) {

while($tmp = $result->fetch\_assoc()) {

$discounts[] = $tmp;

}

$result->close();

}

?>

<?php

$mysqli = mysqli\_init();

if (!$mysqli) {

die('mysqli\_init завершилась провалом');

}

if (!$mysqli->options(MYSQLI\_INIT\_COMMAND, 'SET AUTOCOMMIT = 0')) {

die('Установка MYSQLI\_INIT\_COMMAND завершилась провалом');

}

if (!$mysqli->options(MYSQLI\_OPT\_CONNECT\_TIMEOUT, 5)) {

die('Установка MYSQLI\_OPT\_CONNECT\_TIMEOUT завершилась провалом');

}

if (!$mysqli->real\_connect('localhost', 'root', 'root', 'toy-store')) {

die('Ошибка подключения (' . mysqli\_connect\_errno() . ') ' . mysqli\_connect\_error());

}

?>

<?php

$products = array();

if ($result = $mysqli->query('SELECT \* FROM discounts')) {

while($tmp = $result->fetch\_assoc()) {

$discounts[] = $tmp;

}

$result->close();

}

?>

<?php

$link = mysqli\_connect('localhost', 'root', 'root', 'toy-store');

if(isset($\_POST['but']))

{

$id = $\_POST['idtov'];

$title = $\_POST['titletov'];

$img = $\_POST['imgtov'];

$intro = $\_POST['opistov'];

$price = $\_POST['pricetov'];

$reviews\_count = $\_POST['reviewstov'];

$reviews\_score = $\_POST['scoretov'];

$query = "INSERT INTO `products` (`id`,`title`,`img`,`intro`,`price`,`reviews\_count`,`reviews\_score`) VALUES ('$id','$title','$img','$intro','$price','$reviews\_count','$reviews\_score')";

$result = mysqli\_query($link, $query) or die ("Ошибка " . mysqli\_error($link));

if($result)

{

echo "<span style='color:blue;'>Данные добавлены</span>";

}

mysqli\_close($link);

}

if(isset($\_POST['but1']))

{

$idDel = $\_POST['idtov'];

$query1 = "DELETE FROM `products` WHERE `products`.`id` = $idDel";

$result1 = mysqli\_query($link, $query1) or die ("Ошибка " . mysqli\_error($link));

if($result1)

{

echo "<span style='color:red;'>Запись удалена</span>";

}

mysqli\_close($link);

}

if(isset($\_POST['but2']))

{

$idChange = $\_POST['idtovToChange'];

$titleChange = $\_POST['titletovToChange'];

$imgChange = $\_POST['imgtovToChange'];

$introChange = $\_POST['opistovToChange'];

$priceChange = $\_POST['pricetovToChange'];

$reviews\_countChange = $\_POST['reviewstovToChange'];

$reviews\_scoreChange = $\_POST['scoretovToChange'];

$query2 = "UPDATE `products` SET `title` = '$titleChange', `img` = '$imgChange', `intro` = '$introChange', `price` = '$priceChange', `reviews\_count` = '$reviews\_countChange',`reviews\_score` = '$reviews\_scoreChange' WHERE `products`.`id` = $idChange";

$result2 = mysqli\_query($link, $query2) or die ("Ошибка " . mysqli\_error($link));

if($result2)

{

echo "<span style='color:green;'>Запись отредактирована</span>";

}

mysqli\_close($link);

}

if(isset($\_POST['but3']))

{

$sql = "SELECT \* FROM `products`";

$result = mysqli\_query($link, $sql);

$rows = mysqli\_num\_rows($result);

echo "<table><tr><th>Id |</th><th>Название |</th><th>Изобр. |</th><th>Описание |</th><th>Стоимость |<th>Просмотры |</th> <th>Оценка |</th></th></tr>";

for ($i = 0 ; $i < $rows ; ++$i)

{

$row = mysqli\_fetch\_row($result);

echo "<tr>";

for ($j = 0 ; $j < 7 ; ++$j) echo "<td>$row[$j]</td>";

echo "</tr>";

}

echo "</table>";

}

?>

<?php

session\_start();

if($\_GET['do'] == 'logout'){

unset($\_SESSION['admin']);

session\_destroy();

}

if($\_SESSION['admin'] != "admin"){

header("Location: login.php");

exit;

}

?>

<?php

$mysqli = mysqli\_init();

if (!$mysqli) {

die('mysqli\_init завершилась провалом');

}

if (!$mysqli->options(MYSQLI\_INIT\_COMMAND, 'SET AUTOCOMMIT = 0')) {

die('Установка MYSQLI\_INIT\_COMMAND завершилась провалом');

}

?>

<?php

$products = array();

if ($result = $mysqli->query('SELECT \* FROM products')) {

while($tmp = $result->fetch\_assoc()) {

$products[] = $tmp;

}

$result->close();

}

?>

<?php

session\_start();

$users = 'admin';

$pass = 'rktmai';

if($\_POST['submit']){

if($users == $\_POST['user'] AND $pass == ($\_POST['pass']))

{

$\_SESSION['admin'] = $users;

header("Location: index.php");

exit;

}

$users2 = '123';

$pass2 = '123';

if($\_POST['submit']){

if($users2 == $\_POST['user'] AND $pass2 == ($\_POST['pass']))

{

$\_SESSION['123'] = $users;

header("Location: admin.php");

exit;

}

}

else echo '<p>Логин или пароль неверны!</p>';

}

?>

<?php

$mysqli = mysqli\_init();

if (!$mysqli) {

die('mysqli\_init завершилась провалом');

}

if (!$mysqli->options(MYSQLI\_INIT\_COMMAND, 'SET AUTOCOMMIT = 0')) {

die('Установка MYSQLI\_INIT\_COMMAND завершилась провалом');

}

if (!$mysqli->options(MYSQLI\_OPT\_CONNECT\_TIMEOUT, 5)) {

die('Установка MYSQLI\_OPT\_CONNECT\_TIMEOUT завершилась провалом');

}

if (!$mysqli->real\_connect('localhost', 'root', 'root', 'toy-store')) {

die('Ошибка подключения (' . mysqli\_connect\_errno() . ') ' . mysqli\_connect\_error());

}

}

$result->close();

}

?>

<?php

$mysqli = mysqli\_init();

if (!$mysqli) {

die('mysqli\_init завершилась провалом');

}

if (!$mysqli->real\_connect('localhost', 'root', 'root', 'toy-store')) {

die('Ошибка подключения (' . mysqli\_connect\_errno() . ') ' . mysqli\_connect\_error());

}

?>

<?php

$products = array();

if ($result = $mysqli->query('SELECT \* FROM discounts')) {

while($tmp = $result->fetch\_assoc()) {

$discounts[] = $tmp;

}

$result->close();

}

?>

<?php

$link = mysqli\_connect('localhost', 'root', 'root', 'toy-store');

if(isset($\_POST['but']))

{

$query = "INSERT INTO `products` (`id`,`title`,`img`,`intro`,`price`,`reviews\_count`,`reviews\_score`) VALUES ('$id','$title','$img','$intro','$price','$reviews\_count','$reviews\_score')";

$result = mysqli\_query($link, $query) or die ("Ошибка " . mysqli\_error($link));

if($result)

{

echo "<span style='color:blue;'>Данные добавлены</span>";

}

mysqli\_close($link);

}

if(isset($\_POST['but1']))

{

if($result1)

{

echo "<span style='color:red;'>Запись удалена</span>";

}

mysqli\_close($link);

}

if(isset($\_POST['but2']))

{

$result2 = mysqli\_query($link, $query2) or die ("Ошибка " . mysqli\_error($link));

if($result2)

{

echo "<span style='color:green;'>Запись отредактирована</span>";

}

mysqli\_close($link);

}

if(isset($\_POST['but3']))

{

$sql = "SELECT \* FROM `products`";

$result = mysqli\_query($link, $sql);

$rows = mysqli\_num\_rows($result);

echo "<table><tr><th>Id |</th><th>Название |</th><th>Изобр. |</th><th>Описание |</th><th>Стоимость |<th>Просмотры |</th> <th>Оценка |</th></th></tr>";

for ($i = 0 ; $i < $rows ; ++$i)

{

$row = mysqli\_fetch\_row($result);

echo "<tr>";

for ($j = 0 ; $j < 7 ; ++$j) echo "<td>$row[$j]</td>";

echo "</tr>";

}

echo "</table>";

}

?>

<?php

session\_start();

exit;

}

?>

<?php

<?php

$products = array();

if ($result = $mysqli->query('SELECT \* FROM products')) {

while($tmp = $result->fetch\_assoc()) {

$products[] = $tmp;

}

$result->close();

}

?>

<?php