

**ТЕМА: РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ
РАСПРЕДЕЛЕНИЕ АВТОТРАНСПОРТА СРЕДИ ЗАКАЗЧИКОВ.**

Оглавление

ВВЕДЕНИЕ.....	3
ГЛАВА I. СИСТЕМНЫЙ АНАЛИЗ И ПОСТАНОВКА ЗАДАЧИ. .6	
1.1. Описание автосалона.....	7
1.2. Описание бизнес – процессов учета автомобилей	9
1.3. Постановка задачи	11
ГЛАВА II. ОСНОВНАЯ ЧАСТЬ.....	13
2.1. Создание БД для автоматизированной системы распределение автотранспорта среди заказчиков MySQL.....	14
2.1.1. Создание базы данных MySQL.....	14
2.2. Применение технологии Yii и PHP при создание для автоматизированной системы распределение автотранспорта среди заказчиков	19
2.2.1. Компоненты «Клиента – сервера».....	26
2.3. Обращение к базы данных Создание и Система управление базы данных MySQL.....	28
2.4. Инфологические модели базы данных «Сущность-связи».....	42
2.5 Инструкция для пользователей.....	44
ГЛАВА III. ОХРАНА ТРУДА И ТЕХНИКА БЕЗОПАСНОСТИ 49	
3.1 Пожарная безопасность	50
3.1.1 Основные понятия.....	51
ЗАКЛЮЧЕНИЕ	60
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	61
ПРИЛОЖЕНИЕ.....	62

ВВЕДЕНИЕ

Современный мир информационных технологий трудно представить себе без использования баз данных. Практически все системы в той или иной степени связаны с функциями долговременного хранения и обработки информации. Фактически информация становится фактором, определяющим эффективность любой сферы деятельности. Увеличились информационные потоки и повысились требования к скорости обработки данных, и теперь уже большинство операций не может быть выполнено вручную, они требуют применения наиболее перспективных компьютерных технологий. Любые административные решения требуют четкой и точной оценки текущей ситуации и возможных перспектив ее изменения. И, если раньше в оценке ситуации участвовало несколько десятков факторов, которые могли быть вычислены вручную, то теперь таких факторов сотни и сотни тысяч, и ситуация меняется не в течение года, а через несколько минут, а обоснованность принимаемых решений требуется большая, потому что и реакция на неправильные решения более серьезная, более быстрая и более мощная, чем раньше.

Как правило, базы данных создаются обычно не для решения какой-либо одной задачи для одного пользователя, а для многоцелевого использования. Базы данных отражают определенную часть реального мира. Эта информация должна по возможности фиксироваться в базе данных однократно, и все пользователи, которым эта информация нужна, должны иметь возможность работать с ней. Отсутствие централизованных методов управления доступом к информации послужило причиной разработки систем управления базой данных, а сами хранилища информации, которые работали под управлением данных систем, назывались базами или банками данных.

Процесс автоматизации начинается с середины 70-80 гг. и связан с появлением первой вычислительной техники, которая была не совершенна в современном понимании и требовала специального программного обеспечения и обслуживания – так процесс автоматизации пришел в Узбекистан.

В настоящее время процесс автоматизации затронули не только производственную, техническую и технологическую сферы деятельности человечества, но и информационное пространство; в частности издательства, библиотеки, музеи, информационные центры. Уже сложно представить жизнь общества без таких понятий как интернет, средства телекоммуникации и т.д. Автоматизация больше всего затронула информационную сферу, так как именно информация нуждалась и нуждается в быстром поиске, отборе и хранении, а также в обеспечении доступа к ней.

Основное преимущество автоматизации - это сокращение избыточности хранимых данных, а следовательно, экономия объема используемой памяти, уменьшение затрат на многократные операции обновления избыточных копий и устранение возможности возникновения противоречий из-за хранения в разных местах сведений об одном и том же объекте, увеличение степени достоверности информации и увеличение скорости обработки информации; излишнее количество внутренних промежуточных документов, различных журналов, папок, заявок и т.д., повторное внесение одной и той же информации в различные промежуточные документы. Также значительно сокращает время автоматический поиск информации, который производится из специальных экранных форм, в которых указываются параметры поиска объекта.

Актуальность выпускной квалификационной работы: Для осуществления деятельности в автосалоне используются информационные технологии, автоматизирующие основные процессы: административная, коммерческая автосалон располагает современной материальной базой, включающей вычислительную технику и периферию. Но выдача и распределение автотранспорта осуществляется вручную, что вызывает недовольство некоторых потребителей. По этому я считаю необходимым автоматизировать этот процесс.

Объект и предмет выпускной квалификационной работы: Объектом работы является веб-приложение для автоматизации системы распределение автотранспорта среди заказчиков. Предметом работы выступает технологии

программирования Yii и PHP, а также MySQL, используемые в Разработке автоматизированной системы распределение автотранспорта среди заказчиков.

Цель выпускной квалификационной работы: Целью моей выпускной квалификационной работы является проектирование и разработка качественного и эффективного, максимально оптимизированного и удобного веб-приложения для автоматизации системы распределение автотранспорта среди заказчиков.

ГЛАВА I. СИСТЕМНЫЙ АНАЛИЗ И ПОСТАНОВКА ЗАДАЧИ.

1.1. Описание автосалона

Основным видом деятельности автосалона является реализация автомобилей.

Автосалон - юридическое лицо, имеет зарегистрированный товарный знак, фирменный знак, расчетный и иные счета в рублях и иностранной валюте в учреждениях банков. Автосалон имеет в своем составе администрацию, бухгалтерию, отдел кадров, отдел маркетинга и менеджмента, магазин.

Общее руководство предприятием осуществляет Генеральный директор. В его подчинении находятся Заместитель директора по маркетингу и сбыту и Заместитель директора по экономике и финансам.

Непосредственное управление автосалоном осуществляет Генеральный директор.

Основными подразделениями, обеспечивающими деятельность автосалона, являются:

1. бухгалтерия – прием платежей от покупателей, расчет и выдача зарплаты, налоговая отчетность;
2. отдел менеджмента - оформление заказов, составление отчета о заказах и оплатах этого заказа;
3. отдел маркетинга – определение сбытовой политики с учетом ресурсов и существующей динамики рынка, координация работы всех подразделений;

Технические средства автосалона

Для осуществления деятельности в автосалоне используются информационные технологии, автоматизирующие основные процессы: административная, коммерческая автосалон располагает современной материальной базой, включающей вычислительную технику и периферию..

В автосалоне развернута локальная вычислительная сеть (ЛВС), которая позволяет осуществлять централизованное хранение и обработку информации. Сеть охватывает все отделы.

Инструментальные средства поддержки бизнеса:

На предприятии используются следующие программно-аппаратные средства.

Программные средства:

- Windows 7 Enterprise - операционная система семейства Windows NT корпорации Microsoft.
- Windows Server 2008 — операционная система семейства Windows NT от компании Microsoft, предназначенная для работы на серверах.
- Аппаратные средства:
- В подразделениях предприятия установлены компьютеры со следующей конфигурацией: Intel 3.0 GHz, DDR 512 Mb, VGA 128 Mb
- Периферийные устройства и оргтехника.
- Все компьютеры объединены в единую локальную вычислительную сеть и имеют доступ в Интернет.
- Информационные системы и прикладные программные комплексы:
- MS OFFICE - комплекс программ для подготовки документации.

Описание учета реализации автомобилей в автосалоне

Отдел менеджмента выполняет следующие функции:

- организация принятия заказа от клиента;
- отслеживание состояния заказов;
- составление отчета о заказах и оплатах этого заказа;
- организация работ по рекламе.

Структура отдела менеджмента



Представление аспектов человеческих ресурсов.

- Начальник отдела – осуществляет руководство по развитию и

функционированию, организывает подготовку и проводит внутренние проверки, анализирует их результаты, контролирует выполнение корректирующих мероприятий.

- Менеджер по продажам – осуществляет связь между покупателем и производителем товаров или услуг. Главная задача менеджера по продажам — реализация товара.
- Менеджер по рекламе – осуществляет поиск способов привлечения клиентов, анализ потенциальной клиентской аудитории на активность, ведение переговоров, оформление и заключение договоров, сбор и предоставление статистических данных для рекламодателя.
- Офис-менеджер – выполнение администраторских функций (организация, контроль и др.) заведование хозяйственной частью, АХО секретарские функции (работа с телефоном, почтой, документацией, курьерами), встреча клиентов, информирование о деятельности и услугах компании, составление отчетности о выполненной работе, своевременное предоставление необходимых заявлений, чеков, бланков в бухгалтерию, отдел по работе с персоналом и т.д.

1.2. Описание бизнес – процессов учета автомобилей

Рассмотрим бизнес - процессы, необходимые для учета реализации автомобилей в автосалоне:

1. Заказ автомобиля – после выбора автомобиля оформляется заказ на выбранную модель, подготавливается и отправляется запрос на завод - изготовитель, принимается предоплата и выдается квитанция о предоплате.

2. Прием автомобиля – принятие автомобиля на внутренний учет, проведение предпродажной подготовки и диагностики автомобиля, оповещение покупателя.

3. Реализация автомобиля – осмотр автомобиля покупателем, оформление договора купли-продажи.

4. Регистрация оплаты

5. Формирование отчетных документов:

- Формирование отчета «Прайс-лист» - формирование
- Формирование отчета «Анализ продаж» - формирование списка студентов - должников, сумма долга которых превышает указанную величину (в месяцах) на указанную дату.
- Формирование отчета «Заказы автомобилей» - формирование сведений об операциях, сформированных по студентам за указанный период обучения. Также вывод всей информации о студенте и договоре.
- Формирование отчета «Состояние заказов» - формирование сведений о платежах, сформированных по студентам за указанный период.

Для формализованного описания вышеописанных бизнес-процессов приведены функциональная модель IDEF0 «as-is» в Приложении Б.

Рассмотрим бизнес - правила, необходимые для учета реализации автомобилей в автосалоне:

Бизнес-правила – набор условий, которые управляют деловым событием, чтобы оно происходило так, как нужно для предприятия (или клиента).

- При заказе автомобиля должна быть внесена предоплата, установленная автосалоном
- После постановки автомобиля на внутренний учет, проводится диагностика автомобиля и его предпродажная подготовка, оповещение покупателя
- Заключение договора купли-продажи происходит после полной оплаты стоимости автомобиля

Исследование информационных потоков

В результате анализа деятельности отдела менеджмента был выявлен комплекс информационных потоков.

Входные данные:

- сведения о заказе
- сведения об автомобилях;
- сведения о покупателях

- данные для формирования отчетов;

1.3. Постановка задачи

Исходя из выше перечисленного необходимо определить данные, которые будут использоваться в проектируемой базе данных.

Эти сведения можно разделить на типы:

- сведения о заказе
- сведения об автомобилях;
- сведения о покупателях;
- сведения о продаже;
- данные для формирования отчетов;

Сведения о заказе включают: модель автомобиля, тип кузова, год выпуска, цвет, КПП, объем двигателя, сумма предоплаты.

Сведения о покупателях включают: ФИО покупателя, ИНН, паспортные данные.

Сведения об автомобилях включают: модель автомобиля, тип кузова, год выпуска, цвет, КПП, объем двигателя, номер двигателя.

Сведения о продаже включают: модель автомобиля, ФИО покупателя, ИНН, паспортные данные, цена автомобиля, дата продажи.

Выходные документы должны быть :

- отчет о «Анализ продажах»;
- отчет о«Заказы автомобилей»;
- отчет о «Состояние заказов».

Ожидаемый эффект от внедрения:

- Актуальность – возможность оперативно отражать изменения.
- Полнота информации, то есть предоставление максимально полной информации о составленной смете.
- Удобство – понятный интерфейс.
- Непротиворечивость и достоверность данных.

Консолидация сведений на предприятии.

Автоматизация автосалона по заказу автомобилей и занесение заявки через интернет сайт:

- повышение оперативную выполнению функций;
- обеспечению полноту информацию;
- обеспечение непрерывный оперативный контроли над ходом внесения платежей;
- повышение скорости обработки информации;
- уменьшить число ошибок при учете автомобилей;

ГЛАВА II. ОСНОВНАЯ ЧАСТЬ.

2.1. Создание БД для автоматизированной системы распределение автотранспорта среди заказчиков MySQL

Программное обеспечение MySQL (TM) представляет собой очень быстрый многопоточный, многопользовательский надежный SQL-сервер баз данных (SQL - язык структурированных запросов). Сервер MySQL предназначен как для критических по задачам производственных систем с большой нагрузкой, так и для встраивания в программное обеспечение массового распространения.

Отчеты об ошибках (bugs), а также вопросы и комментарии следует посылать по адресу mysql@lists.mysql.com. See section 1.8.1.3 Как отправлять отчеты об ошибках или проблемах. Для составления отчетов об ошибках следует использовать сценарий `mysqlbug`. В поставках исходного текста сценарий `mysqlbug` находится в директории `scripts`. Если у вас бинарная поставка, то сценарий `mysqlbug` следует искать в директории `'bin'`.

2.1.1. Создание базы данных MySQL

Очень полезным и удобным инструментом, который позволяет создавать базы данных MySQL и работать с ними является `phpMyAdmin`. Он также позволяет тестировать код `php`. Если он еще не установлен на Вашем компьютере, обязательно его установите. Здесь нет ничего сложного.

Входим на страницу администрирования базы данных MySQL. Для того, чтобы зайти на страницу администрирования баз данных, в командной строке Вашего браузера введите следующий адрес: <http://localhost/phpmyadmin/>

В левой колонке находятся имеющиеся базы данных, в центральной части основные настройки (здесь Вы можете изменить язык, вид, кодировку). Верхние вкладки предназначены для различных задач (о них мы еще поговорим в процессе).

Создание базы данных MySQL

Для того, чтобы создать новую базу данных нажмем на верхнюю вкладку «Базы данных» и перед нами на центральном поле откроется список всех имеющихся баз данных MySQL. Нам же нужно создать новую. Для этого в поле

«Создать базу данных» впишем название создаваемой базы и нажмем на кнопку «Создать».

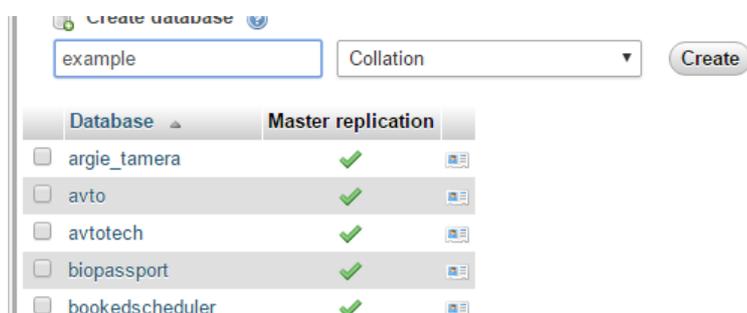


Рис. 2.1. Этап создание БД.

После того, как Вы нажмете на кнопку «Создать» база данных добавиться в список баз данных в панели слева и на центральном поле. Теперь выберете новую базу данных, кликнув по ее названию.

Здесь Вам будет предложено создать таблицу. Давайте создадим таблицу базы данных. Для этого заполните поля «Имя» и «Количество столбцов» и нажмите Ок.

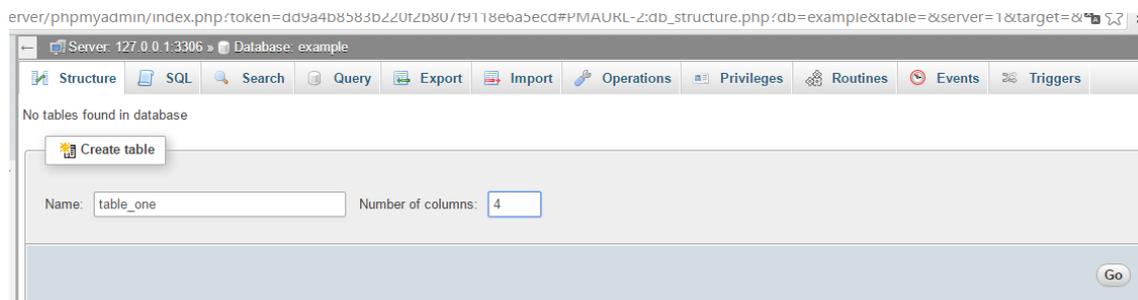


Рис. 2.2. Этап создание таблицы БД.

После этого перед Вами откроется страница для заполнения полей новой таблицы базы данных. Здесь каждому полю нужно присвоить имя, тип хранимых данных, длину (если требуется для данного атрибута) и для такого поля как идентификатор (id) также требуется указать авто-инкремент и первичный ключ. Это должно выглядеть как на скриншоте ниже.

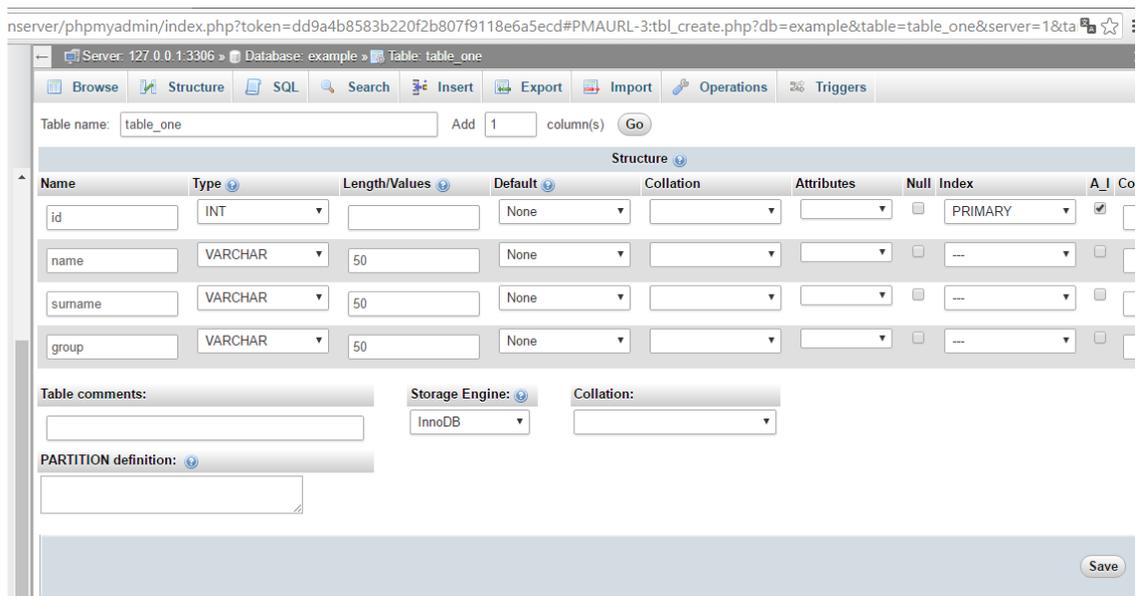


Рис. 2.3. Этап создание столбцов таблицы БД.

Типы данных.

В моем примере нам встретились такие типы данных для таблицы базы как int и varchar.

Int— обозначает, что в данном поле будут храниться целые числа (поле id). Id будет являться идентификатором и для каждого нового элемента он будет разным. При добавлении нового элемента в таблицу, id будет увеличиваться на единицу по сравнению с предыдущим добавленным элементом (именно поэтому поле id должно иметь авто-инкремент (авто приращение) и первичный ключ). С помощью этого поля можно идентифицировать объекты, оно очень поможет нам, когда мы начнем взаимодействовать с базой данных с помощью кода php.

Varchar - тип данных, который может хранить отдельный символ или строку (длиной до 8000 символов). Для таких полей как имя, фамилия, адрес электронной почты он вполне даже подойдет. Еще ему можно задавать длину. В моем примере я задала длину 100, так как все поля в моем примере с этим типом данных не займут более 100 символов.

Существуют и разные другие типы данных, предназначенные для хранения даты, текста и других данных. Вы можете поискать информацию о типах данных в интернете. Также о них я буду рассказывать в следующих уроках.

Далее нажимаем на кнопку «Сохранить» и перед нами созданная таблица, в которой пока нет ни одной записи. Таблица появится у Вас в панели слева и на центральной части экрана. Щелкните по ее имени, чтобы увидеть структуру.

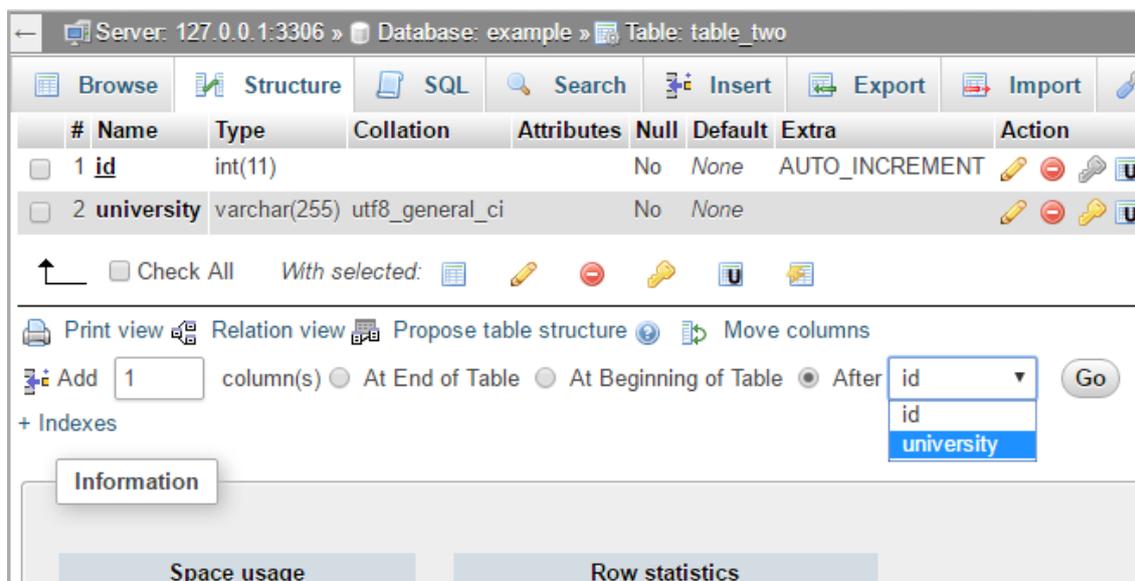


Рис. 2.4. Этап редактирование столбцов таблицы БД.

Здесь можно удалить, изменить и добавить поле. Интерфейс интуитивно понятный и сделать это не составит какого-либо труда.

Сейчас же я покажу как вставить новый элемент в таблицу базы данных.

Для этого кликните по верхней вкладке «Вставить» и перед Вами откроется страница для вставки нового элемента в таблицу базы. Заполните все поля (кроме поля id, оно будет заполняться автоматически) и нажмите на кнопку ОК.

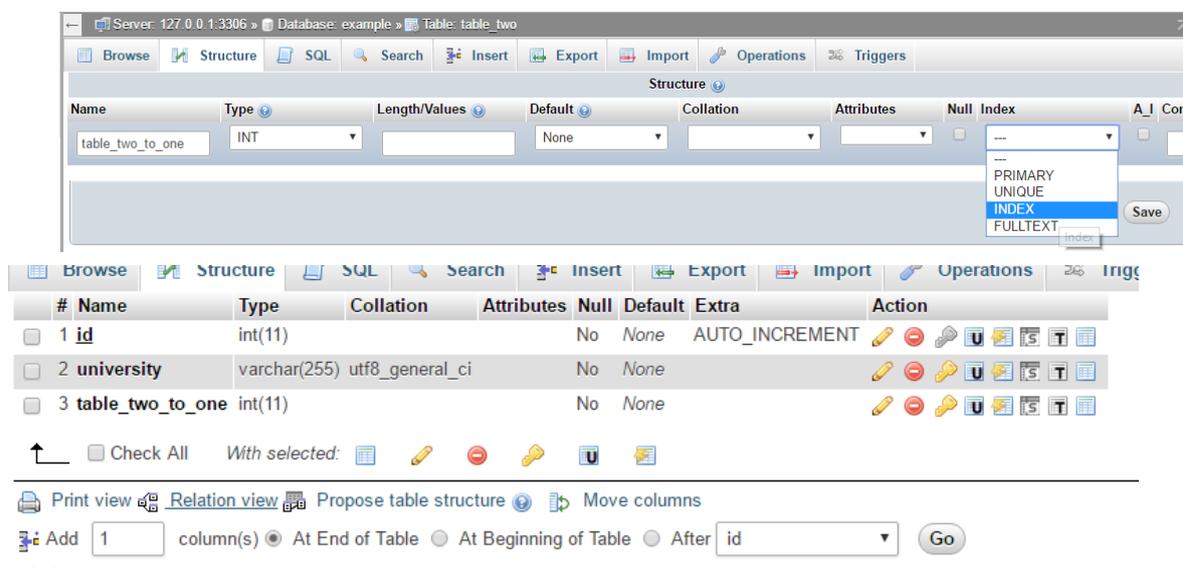


Рис. 2.5. Этап определение типов столбцов таблицы БД.

После того как Вы нажмете Ок, перейдите на вкладку «Обзор» (она находится вверху) и Вы увидите новый добавленный Вами элемент в Вашу таблицу базы данных MySQL.

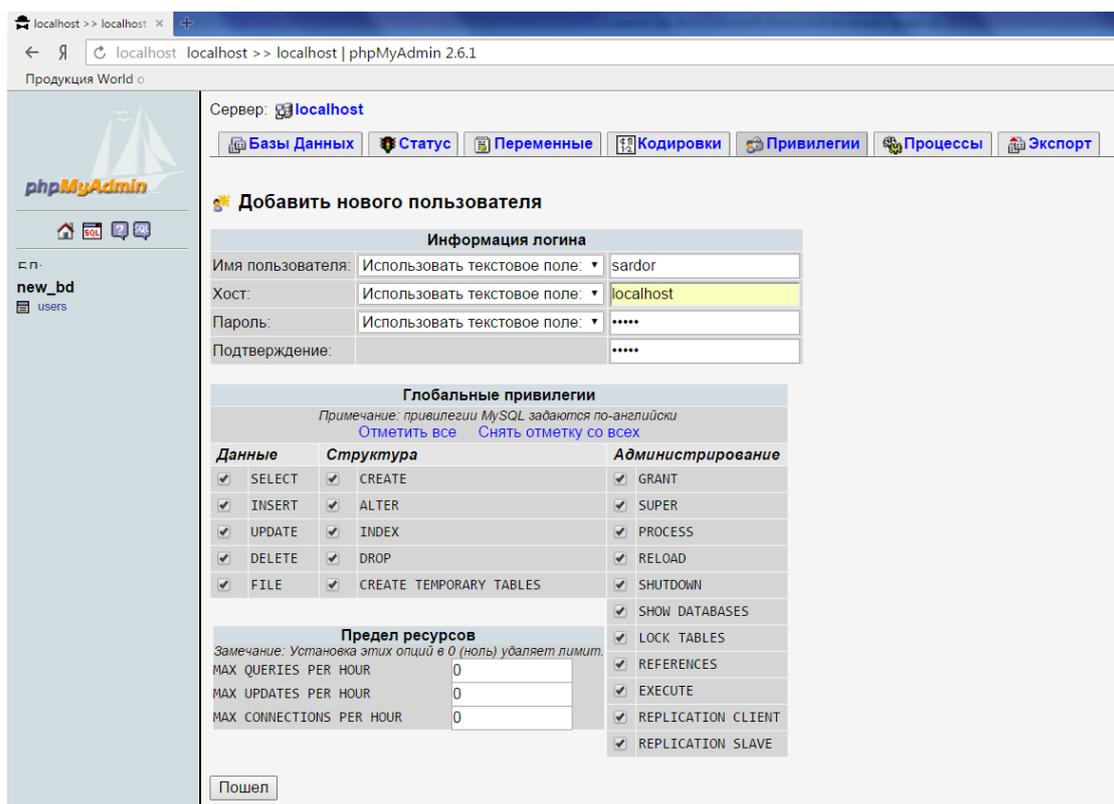
На этой же страничке Вы можете удалить или изменить добавленный элемент из таблицы базы данных.

Теперь я расскажу Вам как создать нового пользователя для базы данных, создать ему имя и пароль и выставить ему определенные привилегии. Информация об имени пользователя и пароле понадобится нам, когда мы будем соединяться с базой при помощи php скрипта.

Итак, создаем нового пользователя для базы данных. Для этого нажимаем вверху на название нашей базы данных, после этого в верхних вкладках появится пункт «Привилегии», нажимаем на него.

Нажимаем «Добавить пользователя». Перед Вами откроется страница с полями, которые нужно заполнить (имя пользователя, хост, пароль и подтверждение пароля). В качестве хоста нужно выбрать локальный хост.

Рис. 2.6. Этап пробного добавление данных в БД.



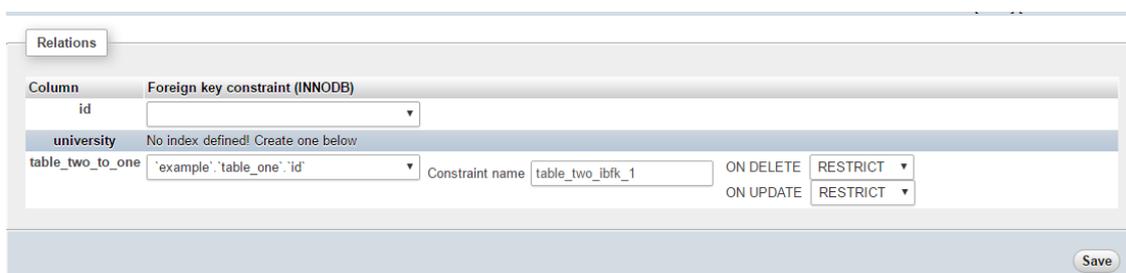
П
осле
чего
нажима
ем
добави
ть
ПОЛЬЗО
вателя
и
НОВЫЙ
ПОЛЬЗО
ватель

будет добавлен. Вы увидите сообщение, что добавлен пользователь к базе данных new_db со всеми привилегиями.

Здесь Вы также можете редактировать привилегии, нажав на редактирование привилегий. Это может понадобиться в том случае, если кто-то еще должен иметь доступ к Вашей базе, но Вы хотите ограничить этого человека в привилегиях (например он не может удалять данные). Тогда Вы создаете нового пользователя для Вашей базы данных, но выставляете ему определенные привилегии.

Наконец, чтобы удалить базу данных нужно снова перейти на вкладку «Базы данных», выбрать базу для удаления и нажать на ссылочку удалить.

Рис. 2.7. Основные операции взаимодействия с базой данных



Здесь мы рассмотрели основные операции взаимодействия с базой данных, через панели администрирования.

2.2. Применение технологии Yii в PHP при создании для автоматизированной системы распределение автотранспорта среди заказчиков

Yii — это высокоэффективный основанный на компонентной структуре PHP-фреймворк для разработки масштабных веб-приложений. Он позволяет максимально применить концепцию повторного использования кода и может существенно ускорить процесс веб-разработки. Название Yii (произносится как Yee или [ji:]) означает простой (easy), эффективный (efficient) и расширяемый (extensible).

Требования

Для запуска веб-приложений, построенных на Yii, вам понадобится веб-сервер с поддержкой PHP версии 5.1.0 или выше.

Для разработчиков, желающих использовать Yii, крайне полезным будет понимание концепции объектно-ориентированного программирования (ООП), так как Yii — это строго объектно-ориентированный фреймворк.

Для чего Yii будет лучшим выбором?

Yii — это фреймворк для веб-программирования общего назначения, который может быть использован для разработки практически любых веб-приложений. Благодаря своей легковесности и наличию продвинутых средств кэширования, Yii особенно подходит для разработки приложений с большим потоком трафика, таких как порталы, форумы, системы управления контентом (CMS), системы электронной коммерции и др.

Yii в сравнении с другими фреймворками

Подобно большинству других PHP-фреймворков, Yii — это MVC-фреймворк.

Превосходство Yii над другими фреймворками заключается в эффективности, широких возможностях и качественной документации. Yii изначально спроектирован очень тщательно для соответствия всем требованиям при разработке серьезных веб-приложений. Yii не является ни побочным продуктом какого-либо проекта, ни сборкой сторонних решений. Он является результатом большого опыта авторов в разработке веб-приложений, а также исследований наиболее популярных веб-фреймворков и приложений.

Создание первого приложения

В этом разделе мы расскажем, как создать наше первое приложение. Для создания нового приложения будем использовать уііс(консольную утилиту), для генерации кода — Gii (мощный веб кодогенератор). Будем считать для удобства, что YiiRoot — это директория, куда установлен Yii, а WebRoot — корневая директория вашего веб-сервера.

Созданное приложение — хорошая отправная точка для добавления необходимого функционала, так как оно уже содержит все необходимые директории и файлы.

Приложение содержит четыре страницы: главную, страницу «о проекте», страницу обратной связи и страницу авторизации. Страница обратной связи содержит форму для отправки вопросов и пожеланий, а страница авторизации позволяет пользователю аутентифицироваться и получить доступ к закрытой части сайта.

Всемирная паутина (англ. WorldWideWeb) — распределённая система, предоставляющая доступ к связанным между собой документам, расположенным на различных компьютерах, подключённых к Интернету. Для обозначения Всемирной паутины также используют слово веб (англ. web «паутина») и аббревиатуру WWW.

Всемирную паутину образуют сотни миллионов веб-серверов. Большинство ресурсов Всемирной паутины основаны на технологии гипертекста. Гипертекстовые документы, размещаемые во Всемирной паутине, называются веб-страницами. Несколько веб-страниц, объединённых общей темой, дизайном, а также связанных между собой ссылками и обычно находящихся на одном и том же веб-сервере, называются веб-сайтом. Для загрузки и просмотра веб-страниц используются специальные программы — браузеры (англ. browser).

Всемирная паутина вызвала настоящую революцию в информационных технологиях и взрыв в развитии Интернета. В повседневной речи, говоря об Интернете, часто имеют в виду именно Всемирную паутину. Однако важно понимать, что это не одно и то же.

Для улучшения визуального восприятия веба стала широко применяться технология CSS, которая позволяет задавать единые стили оформления для множества веб-страниц. Ещё одно нововведение, на которое стоит обратить внимание, — система обозначения ресурсов URN (англ. UniformResourceName).

Популярная концепция развития Всемирной паутины — создание семантической паутины. Семантическая паутина — это надстройка над существующей Всемирной паутиной, которая призвана сделать размещённую в сети информацию более понятной для компьютеров. Семантическая паутина — это концепция сети, в которой каждый ресурс на человеческом языке был бы снабжён описанием, понятным компьютеру. Семантическая паутина открывает доступ к чётко структурированной информации для любых приложений, независимо от платформы и независимо от языков программирования. Программы смогут сами находить нужные ресурсы, обрабатывать информацию, классифицировать данные, выявлять логические связи, делать выводы и даже принимать решения на основе этих выводов. При широком распространении и грамотном внедрении семантическая паутина может вызвать революцию в Интернете. Для создания понятного компьютеру описания ресурса, в семантической паутине используется формат RDF (англ. ResourceDescriptionFramework), который основан на синтаксисе XML и использует идентификаторы URI для обозначения ресурсов. Новинки в этой области — это RDFS (англ. RDFSchema) и SPARQL (англ. ProtocolAndRDFQueryLanguage) (произносится как «спаркл»), новый язык запросов для быстрого доступа к данным RDF.

PHP (рекурсивный акроним словосочетания "PHP: HypertextPreprocessor") - это широко используемый язык программирования общего назначения с открытым исходным кодом. PHP сконструирован специально для ведения Web-разработок и может внедряться в HTML-код.

Значительным отличием PHP от какого-либо кода, выполняющегося на стороне клиента, например, JavaScript, является то, что PHP-скрипты выполняются на сервере. Если бы у вас на сервере был размещен скрипт, подобный вышеприведенному, клиент получил бы только результат выполнения скрипта, причем он не смог бы выяснить, какой именно код выполняется. Вы даже можете сконфигурировать свой сервер таким образом,

чтобы HTML-файлы обрабатывались процессором PHP, так что клиенты даже не смогут узнать, получают ли они обычный HTML-файл или результат выполнения скрипта.

PHP крайне прост для освоения, но вместе с тем способен удовлетворить запросы профессиональных программистов. После того, как вы впервые услышали о PHP, и открыли это руководство, в течение нескольких часов вы уже сможете создавать простые PHP-скрипты.

Хотя PHP, главным образом, предназначен для работы в среде веб-серверов, область его применения не ограничивается только этим. Читайте дальше и не пропустите главу Возможности PHP либо, начните непосредственно с Вводного руководства, если вас интересует исключительно веб-программирование.

Возможности PHP

PHP может все. Главным образом, область применения PHP сфокусирована на написание скриптов, работающих на стороне сервера; таким образом, PHP способен выполнять всё то, что выполняет любая другая программа CGI, например, обрабатывать данных форм, генерировать динамические страницы или отсылать и принимать cookies. Но PHP способен выполнять и множество других задач.

Существуют три основных области, где используется PHP.

- Создание скриптов для выполнения на стороне сервера. PHP наиболее широко используется именно таким образом. Все, что вам понадобится, это парсер PHP (в виде программы CGI или серверного модуля), веб-сервер и браузер. Чтобы вы могли просматривать результаты выполнения PHP-скриптов в браузере, вам нужен работающий веб-сервер и установленный PHP. В случае, если вы просто экспериментируете, вы вполне можете использовать свой домашний компьютер вместо сервера. За более подробными сведениями обратитесь к главе Советы по установке.

- Создание скриптов для выполнения в командной строке. Вы можете создать PHP-скрипт, способный запускаться вне зависимости от

вебсервера и браузера. Все, что вам потребуется - парсер PHP. Такой способ использования PHP идеально подходит для скриптов, которые должны выполняться регулярно, например, с помощью cron (на платформах *nix или Linux) или с помощью планировщика задач (TaskScheduler) на платформах Windows. Эти скрипты также могут быть использованы в задачах простой обработки текстов. За дополнительной информацией обращайтесь к главе Использование PHP в среде командной строки.

- Создание оконных приложений, выполняющихся на стороне клиента. Возможно, PHP является не самым лучшим языком для создания подобных приложений, но, если вы очень хорошо знаете PHP и хотели бы использовать некоторые его возможности в своих клиент-приложениях, вы можете использовать PHP-GTK для создания таких приложений. Подобным образом вы можете создавать и кросс-платформенные приложения. PHP-GTK является расширением PHP и не поставляется вместе с дистрибутивом PHP. Если вы заинтересованы, посетите сайт PHP-GTK.

PHP доступен для большинства операционных систем, включая Linux, многие модификации Unix (такие, как HP-UX, Solaris и OpenBSD), MicrosoftWindows, MacOSX, RISCOS, и многих других. (Совершенно точно, что существует версия PHP для OS/2. Неизвестно, правда, насколько соответствующая нынешним реалиям - Прим.перев.) Также в PHP включена поддержка большинства современных вебсерверов, таких, как Apache, MicrosoftInternetInformationServer, PersonalWebServer, серверов Netscape и iPlanet, сервера OreillyWebsitePro, Caudium, Xitami, OmniHTTPd и многих других. Для большинства серверов PHP поставляется в качестве модуля, для других, поддерживающих стандарт CGI, PHP может функционировать в качестве процессора CGI.

Также в PHP включена поддержка DBX для работы на абстрактном уровне, так что вы можете работать с любой базой данных, использующих DBX. Кроме того, PHP поддерживает ODBC

(OpenDatabaseConnectionstandard), таким образом, вы можете работать с любой базой данных, поддерживающей этот всемирно признанный стандарт.

PHP также поддерживает "общение" с другими сервисами с использованием таких протоколов, как LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (на платформах Windows) и многих других. Кроме того, вы получаете возможность работать с сетевыми сокетами "напрямую". PHP поддерживает стандарт обмена сложными структурами данных WDDX. Обращая внимание на взаимодействие между различными языками, следует упомянуть о поддержке объектов Java и возможности их использования в качестве объектов PHP. Для доступа к удаленным объектам вы можете использовать расширение CORBA.

PHP включает средства обработки текстовой информации, начиная с регулярных выражений Perl или POSIXExtended и заканчивая парсером документов XML. Для парсинга XML в PHP 4 используются стандарты SAX и DOM. Для преобразования документов XML вы можете использовать расширение XSLT. В PHP 5 обработка XML-документов стандартизирована и происходит на базе библиотеки libxml2. Также добавлены два новые расширения для работы с XML: SimpleXML и XMLReader.

Используя PHP в области электронной коммерции, вы обратите внимание на функции осуществления платежей Cybercash, CyberMUT, VeriSignPayflowPro и CCVS.

Последним по порядку, но не по значению, является поддержка многих других расширений, таких, как функции поисковой машины mnoGoSearch, функции IRCGateway, функции для работы со сжатыми файлами (gzip, bz2), функции календарных вычислений, функции перевода.

Как вы видите, этой страницы не хватит для того, чтобы перечислить все, что может предложить вам PHP. Читайте следующую главу, Установка PHP и обратитесь к главе Справочник по функциям за более подробными сведениями о перечисленных выше расширениях.

2.2.1. Компоненты «Клиента – сервера»

Как правило компьютеры и программы, входящие в состав информационной системы, не являются равноправными. Некоторые из них владеют ресурсами (файловая система, процессор, принтер, база данных и т.д.), другие имеют возможность обращаться к этим ресурсам. Компьютер или программа, управляющий ресурсом, называют сервером этого ресурса (файл-сервер, сервер базы данных, вычислительный сервер.). Клиент и сервер какого-либо ресурса могут находиться как на одном компьютере, так и на различных компьютерах, связанных сетью.

В рамках многоуровневого представления вычислительных систем можно выделить три группы функций, ориентированных на решение различных подзадач:

- функции ввода и отображения данных (обеспечивают взаимодействие с пользователем);
- прикладные функции, характерные для данной предметной области;
- функции управления ресурсами (файловой системой, базой данных и т.д.)

Выполнение этих функций в основном обеспечивается программными средствами, которые можно представить в виде взаимосвязанных компонентов.

- компонент представления отвечает за пользовательский интерфейс;
- прикладной компонент реализует алгоритм решения конкретной задачи;
- компонент управления ресурсом обеспечивает доступ к необходимым ресурсам.

Автономная система (компьютер, не подключенный к сети) представляет все эти компоненты как на различных уровнях ОС, служебное ПО и утилиты, прикладное ПО, так и на уровне приложений. Так же и сеть — она представляет все эти компоненты, но, в общем случае,

распределенные между узлами. Задача сводится к обеспечению сетевого взаимодействия между этими компонентами.

Архитектура «клиент-сервер» определяет общие принципы организации взаимодействия в сети, где имеются серверы, узлы-поставщики некоторых специфичных функций (сервисов) и клиенты, потребители этих функций.

Практические реализации такой архитектуры называются клиент-серверными технологиями. Каждая технология определяет собственные или использует имеющиеся правила взаимодействия между клиентом и сервером, которые называются протоколом обмена.

Двухзвенная клиент-серверная архитектура она называется из-за необходимости распределения трех базовых компонентов между двумя узлами (клиентом и сервером).

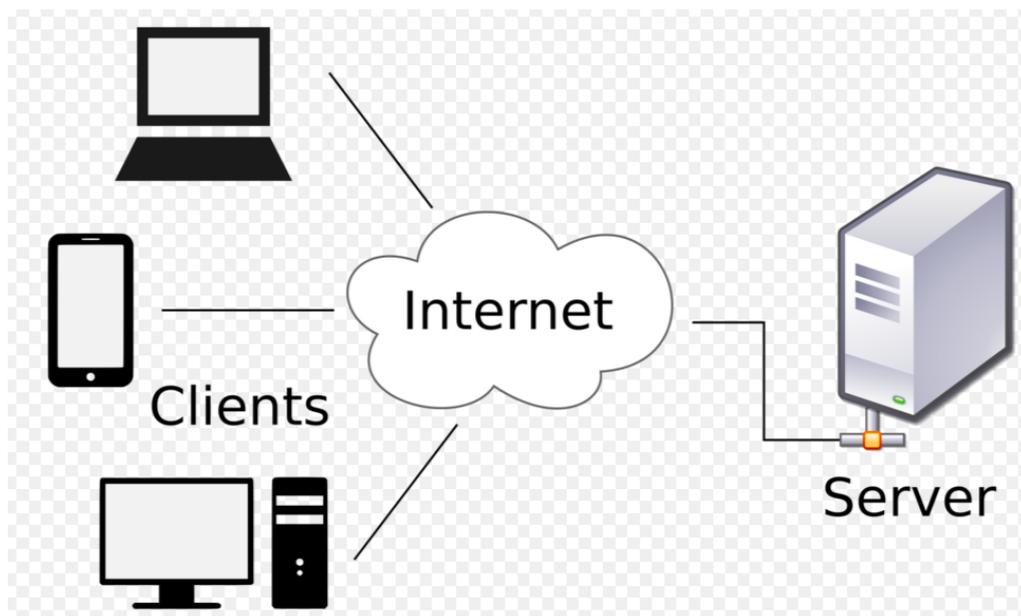


Рис. 2.8. Двухзвенная клиент-серверная архитектура

Двухзвенная архитектура используется в клиент-серверных системах, где сервер отвечает на клиентские запросы напрямую и в полном объеме, при этом используя только собственные ресурсы. Т.е. сервер не вызывает сторонние сетевые приложения и не обращается к сторонним ресурсам для выполнения какой-либо части.

Расположение компонентов на стороне клиента или сервера определяет следующие основные модели их взаимодействия в рамках двухзвенной архитектуры:

сервер терминалов — распределенное представление данных;

файл-сервер — доступ к удаленной базе данных и файловым ресурсам;

сервер БД — удаленное представление данных;

сервер приложений — удаленное приложение.

2.3. Обращение к базе данных Создание и Система управление базы данных MySQL.

Связь PHP с базой данных MySQL. Поработав с phpMyAdmin над созданием базы данных, можно приступить к подключению этой базы данных к внешнему Web-интерфейсу. Чтобы получить доступ к базе данных из Web, используя PHP, надо сделать следующие основные шаги: Подключение к серверу MySQL. Выбор базы данных. Выполнение запроса к базе данных: добавление; удаление; изменение; поиск; сортировка. Получение результата запроса. Отсоединение от базы данных. Для подключения к серверу базы данных в PHP есть функция `mysql_connect()`. Ее аргументы: имя компьютера, имя пользователя и пароль. Эти аргументы можно опустить. По умолчанию имя компьютера = `localhost`, тогда имя пользователя и пароль не требуется. Если PHP используется в сочетании с сервером Apache, то можно воспользоваться функцией `mysql_pconnect()`. В этом случае соединение с сервером не исчезает после завершения работы программы или вызова функции `mysql_close()`. Функции `mysql_connect()` и `mysql_pconnect()` возвращают идентификатор подключения, если все прошло успешно. Например:

После того, как соединение с сервером MySQL установлено, нужно выбрать базу данных. Для этого используется функция `mysql_select_db()`. Ее

аргумент: имя базы данных. Функция возвращает true, если указанная база данных существует и доступ к ней возможен. Например:

После того, как соединение с сервером MySQL установлено, нужно выбрать базу данных. Для этого используется функция `mysql_select_db()`. Ее аргумент: имя базы данных. Функция возвращает true, если указанная база данных существует и доступ к ней возможен. Например:

Для добавления, удаления, изменения и выбора данных нужно сконструировать и выполнить запрос SQL. Для этого в языке PHP существует функция `mysql_query()`. Ее аргумент: строка с запросом. Функция возвращает идентификатор запроса.

При каждом выполнении в таблицу будет добавляться новая запись, содержащая одни и те же данные. Разумеется имеет смысл добавлять в базу данные, введенные пользователем.

Результаты заполнения этой формы передаются в `insert_book.php`.

В введенные строковые данные обработаны функцией `addslashes()`. Эта функция добавляет обратные слешы перед одинарными кавычками (`'`), двойными кавычками (`"`), обратным слешем (`\`) и null-байтом. Дело в том, что по требованиям синтаксиса запросов баз данных такие символы должны заключаться в кавычки. Для определения количества записей в результате запроса используется функция `mysql_num_rows()`. Все записи результата запроса можно просмотреть в цикле. Перед этим с помощью функции `mysql_fetch_array()` для каждой записи получают ассоциативный массив.

Результаты заполнения этой формы передаются в `search_book.php`.

Итак, как работает архитектура Web-баз данных: Web-браузер пользователя выдает HTTP-запрос определенной Web-страницы. Например, пользователь, используя HTML-форму, ищет все книги о PHP. Страница обработки формы называется `search_book.php`. Web-сервер принимает запрос на `search_book.php`, извлекает этот файл и передает на обработку механизму PHP. PHP выполняет соединение с MySQL-сервером и отправляет запрос. Сервер принимает запрос к базе данных, обрабатывает его и отправляет

результат (список книг) обратно механизму PHP. Механизм PHP завершает выполнение сценария, форматирует результат запроса в HTML. После этого результат в виде HTML возвращается Web-серверу.

SELECTFOR UPDATE Если Вы запускаете несколько процессов, которые делают select запрос к одной и той же таблице, то они могут выбрать одну и ту же запись одновременно. Чтобы избежать вышеупомянутой ситуации необходимо выполнить не просто SELECT запрос, а его расширенную версию, о которой многие и не подозревают: **SELECTFOR UPDATE**. Таким образом, при выполнении данного запроса, все затронутые записи в базе данных будут заблокированы до завершения сеанса работы с БД или до момента обновления данных записей. Другой скрипт не сможет выбрать заблокированные записи до тех пор, пока не наступит одно из упомянутых условий. Однако не всё так просто. Вам нужно выполнить ещё несколько условий. Во-первых, ваша таблица должна быть создана на основе архитектуры InnoDB. В противном случае блокировка просто не будет срабатывать. Во-вторых, перед выполнением выборки необходимо отключить авто-коммит запроса. Т.е. другими словами автоматическое выполнение запроса. После того как вы укажете UPDATE запрос, необходимо будет ещё раз обратиться к базе и за коммитить изменения с помощью команды COMMIT:

Представьте что у нас есть 2 записи в таблице. Мы запускаем два параллельных скрипта с данным содержанием. В результате, первый процесс, который сможет достучаться до базы сделает выборку записи и сразу же заменит значение в поле locked на. В результате второй скрипт проигнорирует первую запись. В данном случае, даже если скрипты достучались бы до БД одновременно, второй скрипт всё равно не смог бы выбрать первую запись, т.к. при выборке первым процессом она будет заблокирована.

Реализация операций CRUD

А теперь самое веселое. Мы бы хотели добавить операции CRUD (создание, чтение, обновление и удаление) для только что созданной таблицы `tbl_user` — это часто необходимо при создании реальных приложений. Вместо написания кода мы воспользуемся веб-кодогенератором Gii.

Программное обеспечение MySQL (TM) представляет собой очень быстрый многопоточный, многопользовательский надежный SQL-сервер баз данных (SQL - язык структурированных запросов). Сервер MySQL предназначен как для критических по задачам производственных систем с большой нагрузкой, так и для встраивания в программное обеспечение массового распространения.

MySQL - это торговая марка MySQL AB.

Программное обеспечение MySQL имеет двойное лицензирование. Это означает, что пользователи могут выбирать, использовать ли ПО MySQL бесплатно по общедоступной лицензии GNUGeneralPublicLicense (GPL) или приобрести одну из стандартных коммерческих лицензий MySQLAB.

Для получения самой свежей информации о программном обеспечении MySQL обращайтесь на веб-сайт MySQL

Что важно:

Отчеты об ошибках (bugs), а также вопросы и комментарии следует посылать по адресу `mysql@lists.mysql.com`. See section 1.8.1.3 Как отправлять отчеты об ошибках или проблемах. Для составления отчетов об ошибках следует использовать сценарий `mysqlbug`. В поставках исходного текста сценарий `mysqlbug` находится в директории `scripts`. Если у вас бинарная поставка, то сценарий `mysqlbug` следует искать в директории `'bin'`.

Если вы обнаружите существенную ошибку, относящуюся к безопасности в сервере MySQL, следует сообщить об этом по адресу: `security@mysql.com`.

Оптимизация в MySQL

Оптимизация - сложная задача, потому что она, в конечном счете, требует понимания системы в целом. В отдельных случаях для выполнения

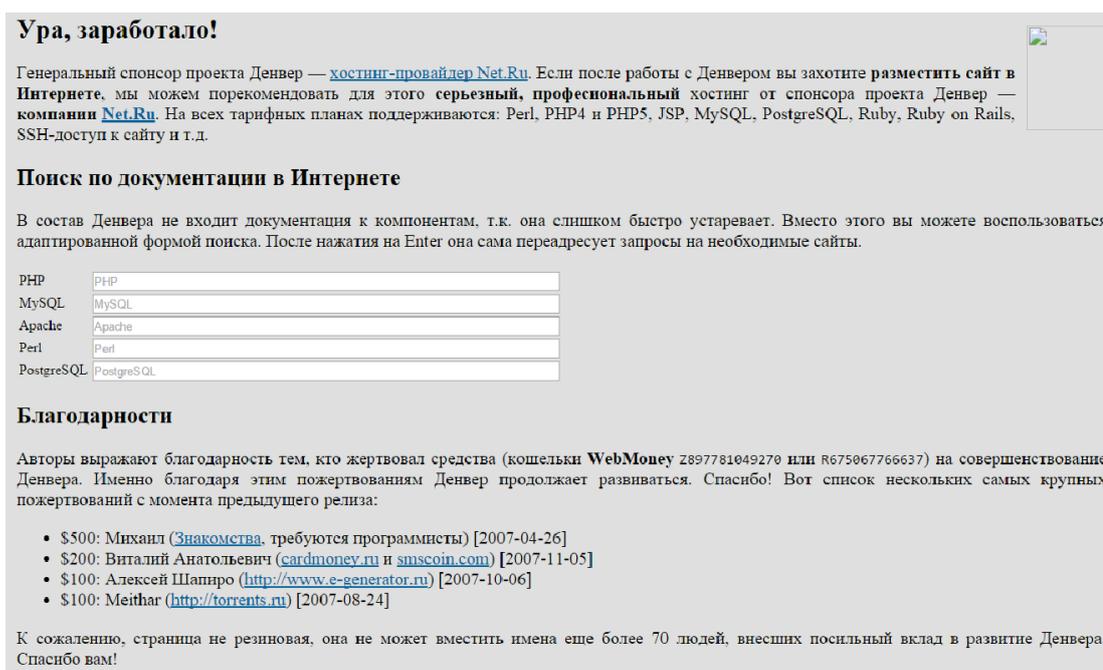
локальной оптимизации достаточно знать систему или приложение не в полном объеме, однако чтобы сделать систему более оптимальной, нужно разбираться в том, как она устроена.

В этой главе будут рассмотрены различные способы оптимизации MySQL и представлены некоторые примеры ее выполнения. Не следует забывать, однако, что всегда можно найти некоторые дополнительные возможности сделать систему еще быстрее (хотя каждый следующий шаг в этом направлении будет даваться все труднее и труднее).

Создание БД

Очень полезным и удобным инструментом, который позволяет создавать базы данных MySQL и работать с ними является Denwer. Он также позволяет тестировать код php. Если он еще не установлен на Вашем компьютере, обязательно его установите. Здесь нет ничего сложного. Все что от Вас потребуется это перейти по ссылке на сайт Denwer , скачать Денвер на Ваш компьютер. На этой же странице Вы найдете подробную инструкцию по установке Денвера, которую я не вижу смысла переписывать в эту статью.

После удачной установки Денвера (когда Вы увидели надпись «Ура, заработало!») начнем работу с администраторской страницей MySQL.



Ура, заработало!

Генеральный спонсор проекта Денвер — [хостинг-провайдер Net.Ru](#). Если после работы с Денвером вы захотите **разместить сайт в Интернете**, мы можем порекомендовать для этого **серьезный, профессиональный хостинг** от спонсора проекта Денвер — компании [Net.Ru](#). На всех тарифных планах поддерживаются: Perl, PHP4 и PHP5, JSP, MySQL, PostgreSQL, Ruby, Ruby on Rails, SSH-доступ к сайту и т.д.

Поиск по документации в Интернете

В состав Денвера не входит документация к компонентам, т.к. она слишком быстро устаревает. Вместо этого вы можете воспользоваться адаптированной формой поиска. После нажатия на Enter она сама переадресует запросы на необходимые сайты.

PHP	<input type="text" value="PHP"/>
MySQL	<input type="text" value="MySQL"/>
Apache	<input type="text" value="Apache"/>
Perl	<input type="text" value="Perl"/>
PostgreSQL	<input type="text" value="PostgreSQL"/>

Благодарности

Авторы выражают благодарность тем, кто жертвовал средства (кошельки **WebMoney** Z897781049270 или R675067766637) на совершенствование Денвера. Именно благодаря этим пожертвованиям Денвер продолжает развиваться. Спасибо! Вот список нескольких самых крупных пожертвований с момента предыдущего релиза:

- \$500: Михаил ([Знакомства](#), требуются программисты) [2007-04-26]
- \$200: Виталий Анатольевич ([cardmoney.ru](#) и [smscoin.com](#)) [2007-11-05]
- \$100: Алексей Шапиро (<http://www.e-generator.ru>) [2007-10-06]
- \$100: Meithar (<http://torrents.ru>) [2007-08-24]

К сожалению, страница не резиновая, она не может вместить имена еще более 70 людей, внесших посильный вклад в развитие Денвера. Спасибо вам!

Рис. 2.9 Вид окна PhPMyAdmin

Входим на страницу администрирования базы данных MySQL. Для того, чтобы зайти на страницу администрирования баз данных, в командной строке Вашего браузера введите следующий адрес: localhost/tools/phpmyadmin/.

В левой колонке находятся имеющиеся базы данных, в центральной части основные настройки (здесь Вы можете изменить язык, вид, кодировку). Верхние вкладки предназначены для различных задач (о них мы еще поговорим в процессе).

Создание базы данных MySQL

Для того, чтобы создать новую базу данных нажмем на верхнюю вкладку «Базы данных» и перед нами на центральном поле откроется список всех имеющихся баз данных MySQL. Нам же нужно создать новую. Для этого в поле «Создать базу данных» впишем название создаваемой базы и нажмем на кнопку «Создать».

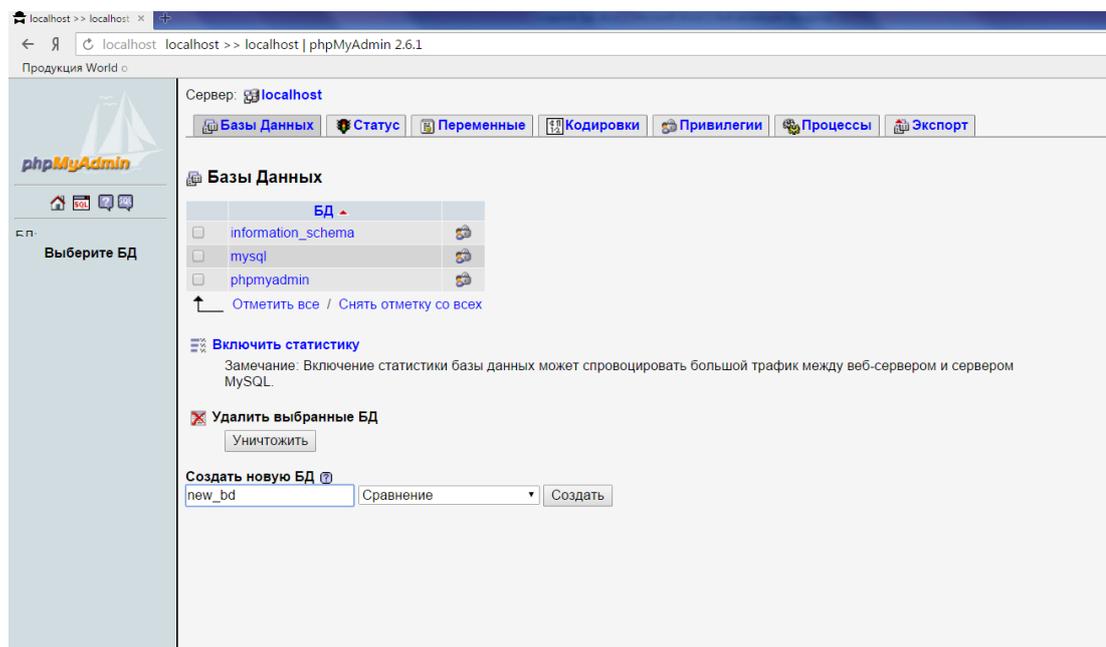


Рис. 2.10 Этап создание БД для Автоматизированной системы паспортного стола

После того, как Вы нажмете на кнопку «Создать» база данных добавиться в список баз данных в панели слева и на центральном поле. Теперь выберете новую базу данных, кликнув по ее названию.

Здесь Вам будет предложено создать таблицу. Давайте создадим таблицу базы данных. Для этого заполните поля «Имя» и «Количество столбцов» и нажмите Ок.

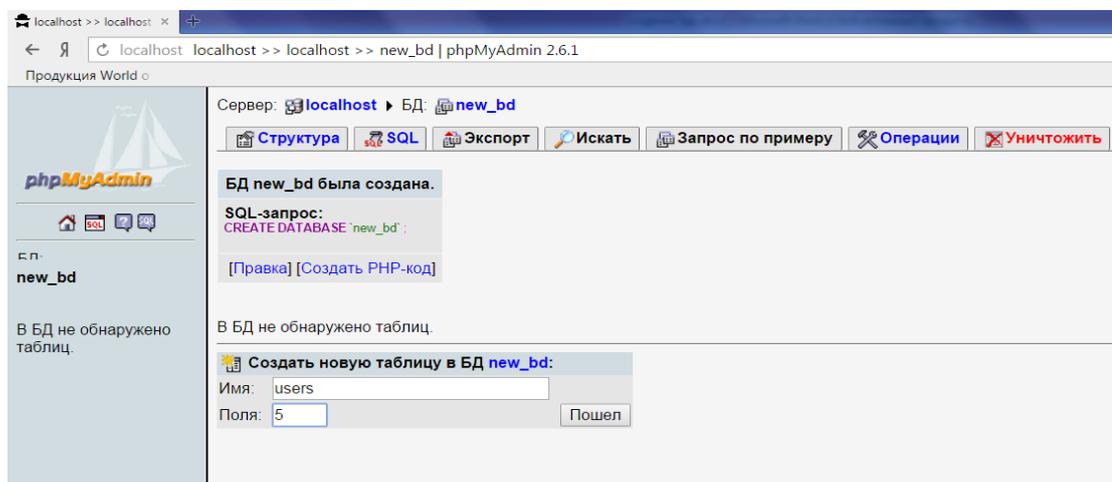


Рис. 2.11 Этап распределение таблиц Автоматизированной системы паспортного стола

После этого перед Вами откроется страница для заполнения полей новой таблицы базы данных. Здесь каждому полю нужно присвоить имя, тип хранимых данных, длину (если требуется для данного атрибута) и для такого поля как идентификатор (id) также требуется указать авто-инкремент и первичный ключ. Это должно выглядеть как на скриншоте ниже.

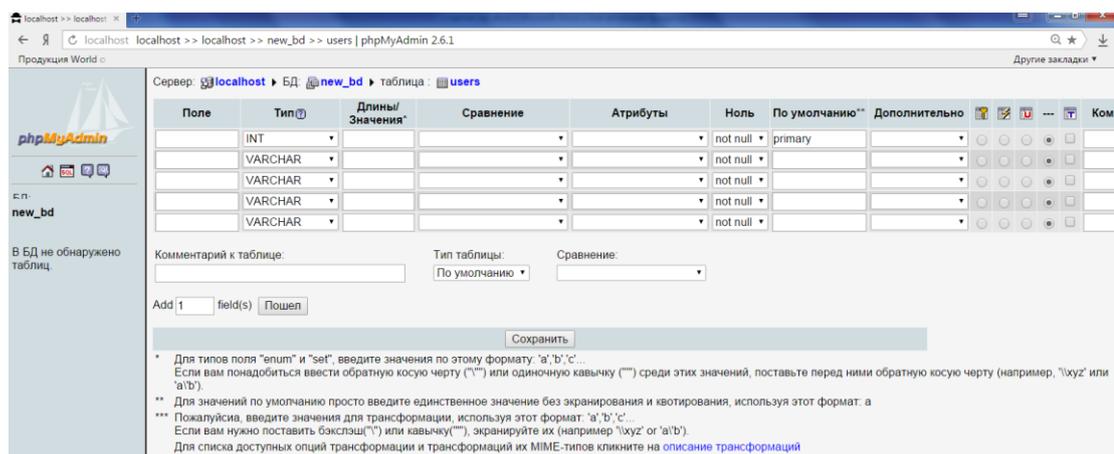


Рис. 2.12 Этап обозначение столбцов таблицы Автоматизированной системы паспортного стола

Типы данных.

В моем примере нам встретились такие типы данных для таблицы базы как int и varchar.

Int – обозначает, что в данном поле будут храниться целые числа (поле `id`). `Id` будет являться идентификатором и для каждого нового элемента он будет разным. При добавлении нового элемента в таблицу, `id` будет увеличиваться на единицу по сравнению с предыдущим добавленным элементом (именно поэтому поле `id` должно иметь авто-инкремент (автоприращение) и первичный ключ). С помощью этого поля можно идентифицировать объекты, оно очень поможет нам, когда мы начнем взаимодействовать с базой данных с помощью кода `php`.

Varchar - тип данных, который может хранить отдельный символ или строку (длиной до 8000 символов). Для таких полей как имя, фамилия, адрес электронной почты он вполне даже подойдет. Еще ему можно задавать длину. В моем примере я задала длину 100, так как все поля в моем примере с этим типом данных не займут более 100 символов.

Существуют и разные другие типы данных, предназначенные для хранения даты, текста и других данных. Вы можете поискать информацию о типах данных в интернете. Также о них я буду рассказывать в следующих уроках.

Далее нажимаем на кнопку «Сохранить» и перед нами созданная таблица, в которой пока нет ни одной записи. Таблица появится у Вас в панели слева и на центральной части экрана. Щелкните по ее имени, чтобы увидеть структуру.

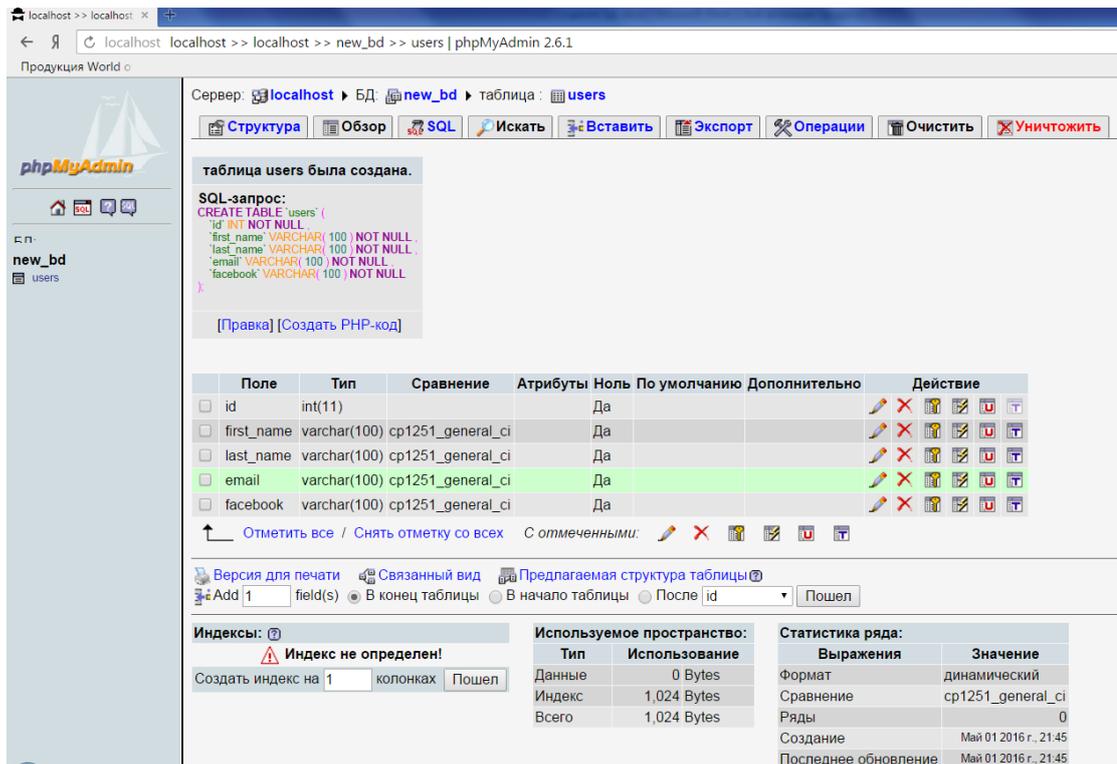


Рис. 2.13 Этап редактирование столбцов таблицы

Здесь можно удалить, изменить и добавить поле. Интерфейс интуитивно понятный и сделать это не составит какого-либо труда.

Сейчас же я покажу как вставить новый элемент в таблицу базы данных.

Для этого кликните по верхней вкладке «Вставить» и перед Вами откроется страница для вставки нового элемента в таблицу базы. Заполните все поля (кроме поля id, оно будет заполняться автоматически) и нажмите на кнопку ОК.

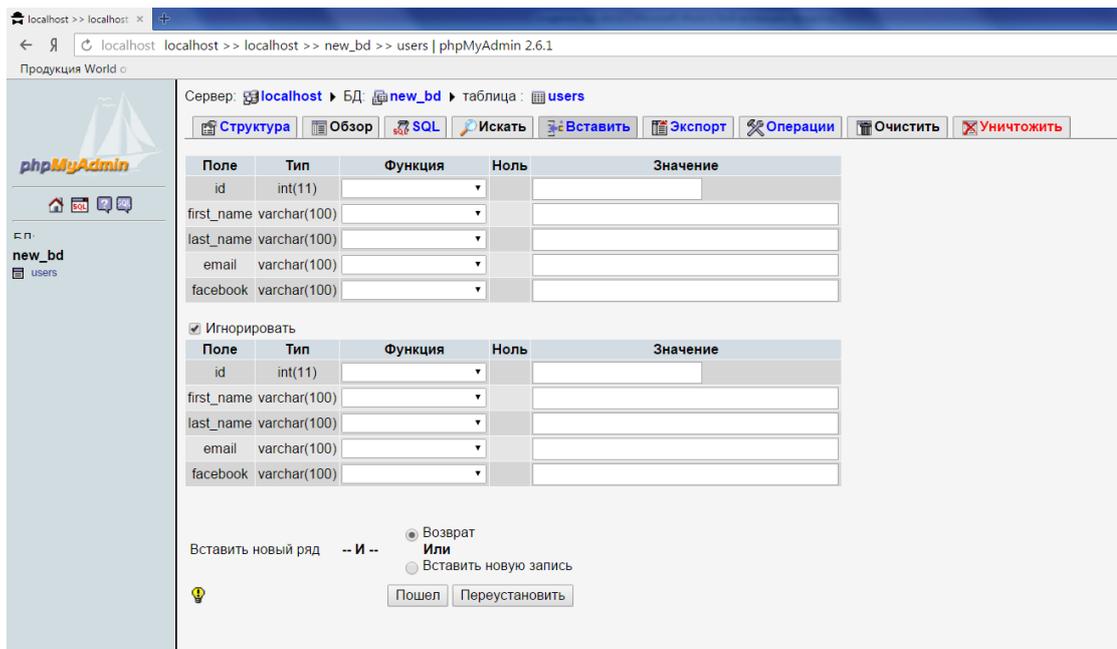


Рис. 2.14 Вид созданных столбцов таблицы

После того как Вы нажмете Ок, перейдите на вкладку «Обзор» (она находится сверху) и Вы увидите новый добавленный Вами элемент в Вашу таблицу базы данных MySQL.

На этой же страничке Вы можете удалить или изменить добавленный элемент из таблицы базы данных.

Теперь я расскажу Вам как создать нового пользователя для базы данных, создать ему имя и пароль и выставить ему определенные привилегии. Информация об имени пользователя и пароле понадобится нам, когда мы будем соединяться с базой при помощи php скрипта.

Итак, создаем нового пользователя для базы данных. Для этого нажимаем сверху на название нашей базы данных, после этого в верхних вкладках появится пункт «Привилегии», нажимаем на него.

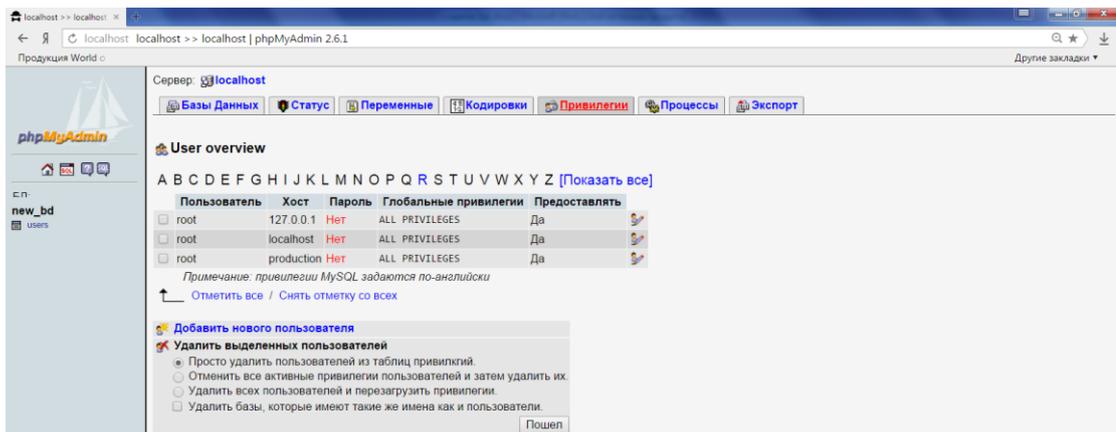


Рис. 2.15 Этап редактирование столбцов таблицы

Нажимаем «Добавить пользователя». Перед Вами откроется страница с полями, которые нужно заполнить (имя пользователя, хост, пароль и подтверждение пароля). В качестве хоста нужно выбрать локальный хост.

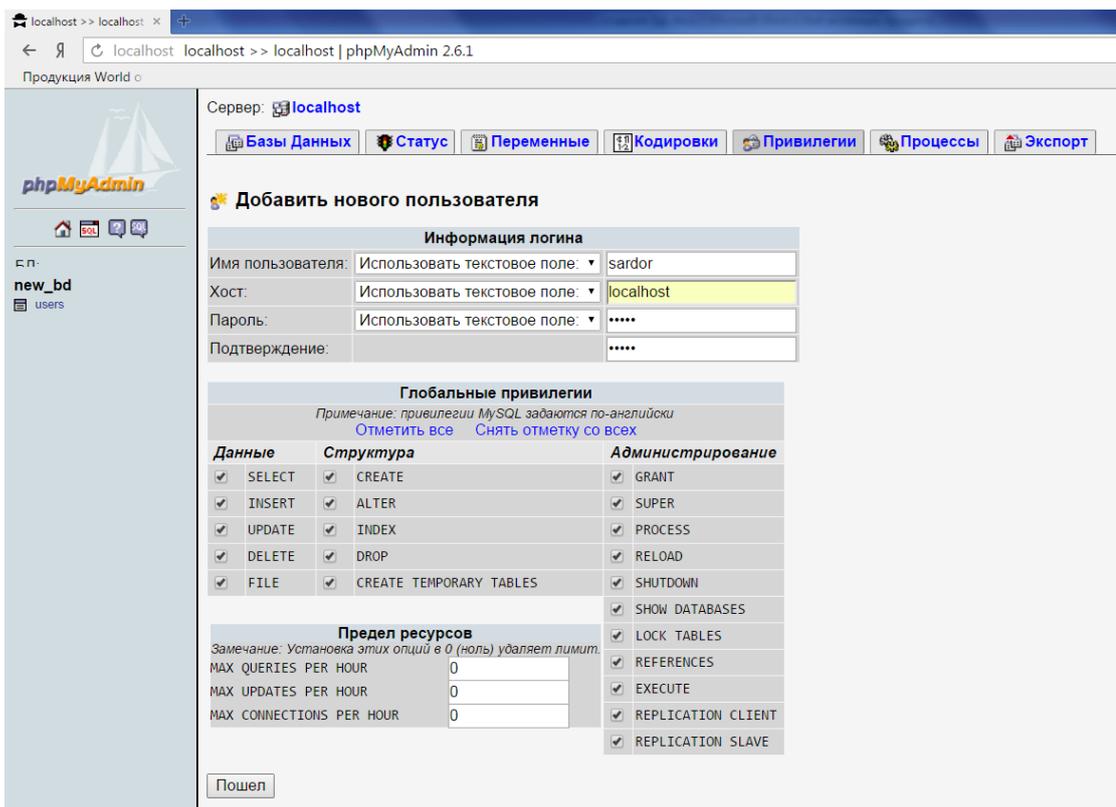


Рис. 2.16 Этап добавление данных

После чего нажимаем добавить пользователя и новый пользователь будет добавлен. Вы увидите сообщение, что добавлен пользователь к базе данных new_db со всеми привилегиями.

Здесь Вы также можете редактировать привилегии, нажав на редактирование привилегий. Это может понадобиться в том случае, если кто-то еще должен иметь доступ к Вашей базе, но Вы хотите ограничить этого человека в привилегиях (например он не может удалять данные). Тогда Вы создаете нового пользователя для Вашей базы данных, но выставляете ему определенные привилегии.

Наконец, чтобы удалить базу данных нужно снова перейти на вкладку «Базы данных», выбрать базу для удаления и нажать на ссылочку удалить.

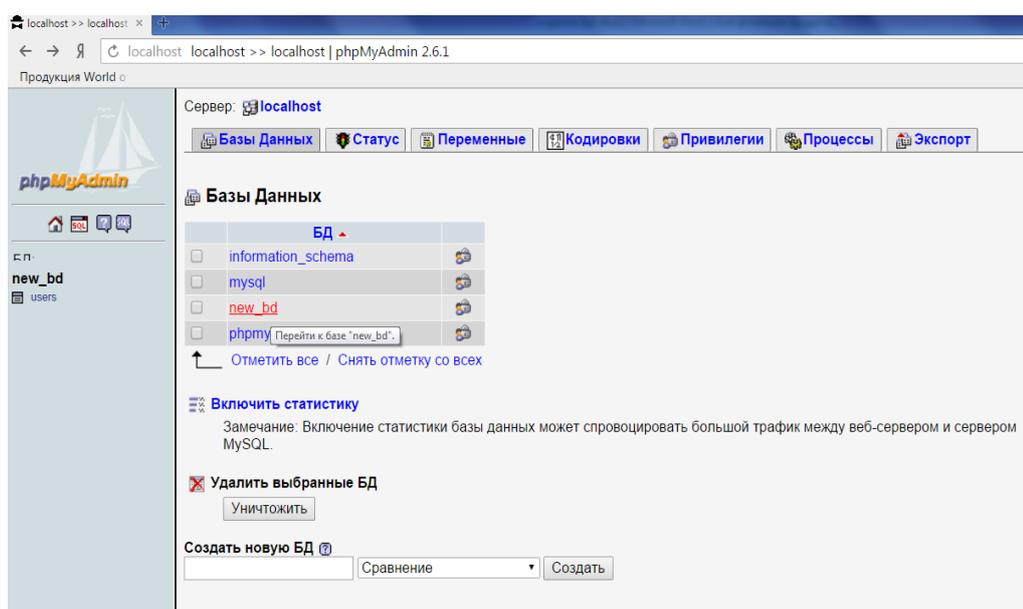


Рис. 2.17 Этап редактирование БД

Здесь мы рассмотрели основные операции взаимодействия с базой данных, через панели администрирования.

Система управления базы данных

MySQL – свободно-распространяемая система управления базами данных, доступная для множества платформ. Изначально была разработана Майклом Видениусом и принадлежала специально созданной для развития этой СУБД компании MySQL AB. В 2008 году компания была приобретена sunMicrosystems, а с 2010 – стала принадлежать Oracle после того как последняя поглотила Sun. Несмотря на эти коммерческие метаморфозы, MySQL остается самой популярной СУБД в мире программного обеспечения с открытым исходным кодом.

MySQL имеет двойное лицензирование:

- GNU GPL (лицензия на свободное ПО);
- коммерческая лицензия, позволяющая использовать MySQL, не открывая своего исходного кода.

В общем случае у MySQL две основные версии:

- бесплатная – MySQLCommunityServer, она наиболее популярна: предоставляется хостинг-компаниям для веб-сайтов, входит в состав многочисленных Linux-дистрибутивов;
- коммерческая – MySQLEnterprise, включает в себя ряд дополнительных инструментов для работы с БД, а также включает техническую поддержку.

MySQL – это быстрая, надежная, открыто распространяемая СУБД. MySQL, как и многие другие СУБД, функционирует по модели «клиент/сервер». На рисунке 2.21 изображена схема передачи информации между компьютерами клиента и жестким диском сервера.



Рис. 2.18– Схема передачи данных в архитектуре «клиент/сервер»

MySQL – это система управления реляционными базами данных. В системе может быть создана одна или несколько БД.

База данных представляет собой совокупность информации, организованной в виде множества. Каждое множество содержит записи

унифицированного вида. Сами записи состоят из полей. На жестком диске вся база данных может находиться в одном файле. Таблицы могут храниться и описываться по-разному, в зависимости от типа, который вы для нее установите при создании. Более подробно существующие типы мы рассмотрим далее.

Клиентская программа MySQL представляет собой утилиту командной строки. Эта программа подключается к серверу по сети. Команды выполняемые сервером, обычно связаны с чтением и записью данных на жестком диске.

Базовый интерфейс работы с СУБД – терминал, хотя для работы с ней существует множество графических приложений. MySQL взаимодействует с базой данных на языке SQL.

Основные характеристики СУБД:

- MySQL – это система управления реляционными БД;
- программное обеспечение этой СУБД – это ПО с открытым кодом, т.е. применять и модифицировать его может любой желающий;
- технические возможности - ПО MySQL является системой клиент-сервер, которая содержит многопоточный SQL-сервер, обеспечивающий поддержку различных вычислительных машин БД, а также несколько различных клиентских программ и библиотек, средства администрирования и широкий спектр программных интерфейсов (API);
- система безопасности основана на привилегиях и паролях с возможностью верификации с удаленного компьютера, за счет чего обеспечивается гибкость и безопасность. Пароли при передаче по сети при соединении с сервером шифруются;
- вместимость данных – максимальный размер таблицы доведен до 8 миллионов терабайт, однако каждая ОС имеет свои ограничения по размерам файлам. По умолчанию MySQL-таблицы имеют максимальный размер около 4 Гб.

2.4.Инфологические модели базы данных «Сущность-связи»

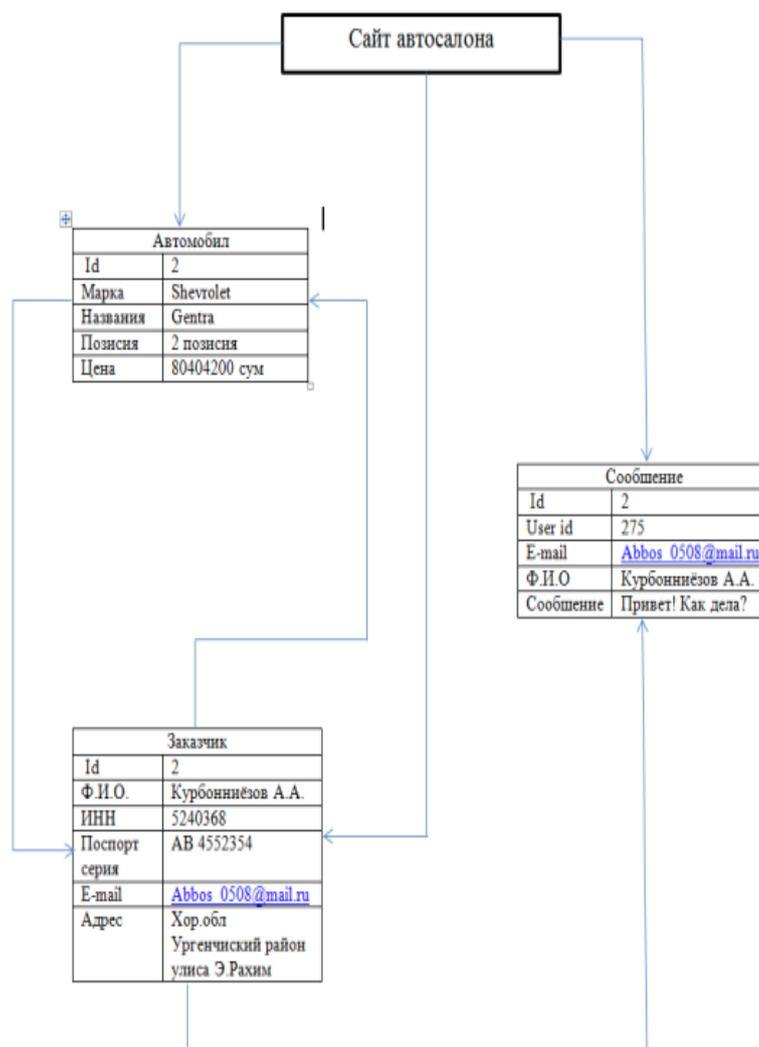


Рис 2.19. Инфологические модели базы данных «Сущность-связи»

Цель инфологического моделирования – обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных. Поэтому инфологическую модель данных пытаются строить по аналогии с естественным языком (последний не может быть использован в чистом виде из-за сложности компьютерной обработки текстов и неоднозначности любого естественного языка). Основными конструктивными элементами инфологических моделей являются сущности, связи между ними и их свойства.

Сущность – любой различимый объект (объект, который мы можем отличить от другого), информацию о котором необходимо хранить в базе данных. Сущностями могут быть люди, места, самолеты, рейсы, вкус, цвет и т.д.

Необходимо различать такие понятия, как *тип сущности* и *экземпляр сущности*. Понятие тип сущности относится к набору однородных личностей, предметов, событий или идей, выступающих как целое. Экземпляр сущности относится к конкретной вещи в наборе.

Атрибут – поименованная характеристика сущности. Его наименование должно быть уникальным для конкретного типа сущности, но может быть одинаковым для различного типа сущностей. Атрибуты используются для определения того, какая информация должна быть собрана о сущности. Абсолютное различие между типами сущностей и атрибутами отсутствует. Атрибут является таковым только в связи с типом сущности. В другом контексте атрибут может выступать как самостоятельная сущность.

Ключ – минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности. Минимальность означает, что исключение из набора любого атрибута не позволяет идентифицировать сущность по оставшимся.

Связь – ассоциирование двух или более сущностей. Если бы назначением базы данных было только хранение отдельных, не связанных между собой данных, то ее структура могла бы быть очень простой. Однако одно из основных требований к организации базы данных – это обеспечение возможности отыскания одних сущностей по значениям других, для чего необходимо установить между ними определенные связи. А так как в реальных базах данных нередко содержатся сотни или даже тысячи сущностей, то теоретически между ними может быть установлено более миллиона связей. Наличие такого множества связей и определяет сложность инфологических моделей.

При построении инфологических моделей можно использовать язык *ER-диаграмм* (от англ. Entity-Relationship, т.е. сущность-связь). В них сущности изображаются помеченными прямоугольниками, ассоциации – помеченными ромбами или шестиугольниками, атрибуты – помеченными овалами, а связи между ними – ненаправленными ребрами, над которыми может проставляться

степень связи (1 или буква, заменяющая слово "много") и необходимое пояснение.

2.5 Инструкция для пользователей

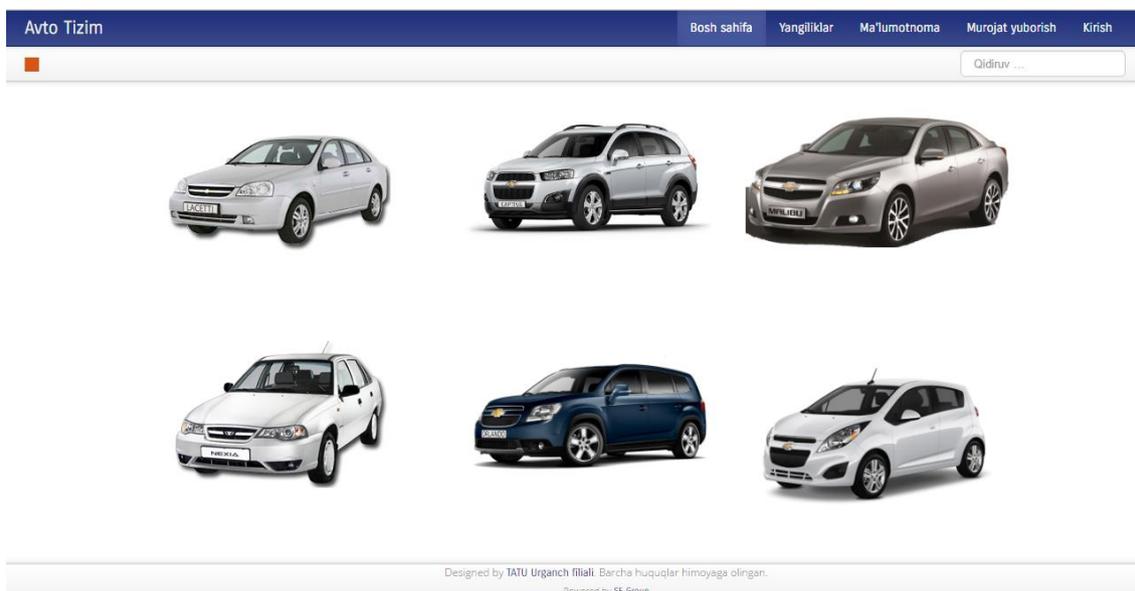


Рис 2.20. Знакомство в web приложения где можно выбрать автомобиля

Avto Tizim Bosh sahifa Yangiliklar Ma'lumotnoma Murojat yuborish Kirish

Qidiruv ...

Buyurtmalar ██████████

Salondagi mashinalar ro'yhati ██████████

Administrator uchun ro'yhat Chop qilish

- 1 Murojatlar
- 4 Ro'yhatdan o'lganlar

ID: 11



Mashina markasi : Lacetti
To'liq ma'lumot:

To'g'ri mutanosiblik, silliq forma, bo'rttirib ishlangan chiziqlar bir-biriga hamohang tarzda savlatli interyer va eksteryerga ega avtomobil. Lacetti zamonaviy va shinam saloniga besh nafar insomni juda qulay joylashtira oladi. Uskunalar paneli haddidan oshmaydigan vilne-xil uslublari ajralib turadi. Xrom bilan ishlov berilgan qismlar va qurilmalar mukammal joylashtirilgan. Avtomobilning yurish qismi jiddiy sinovlardan o'tkazilgan hamda mukammal darajaga yetkazilgan. Lacetti kuzovining zamonaviy dizayni Italiyaning mashhur Pininfarina avtomobil atelyesi mehnatining mahsulidir. Bodomsimon oldchiroqlar va uzun ohista chiziqlar kishini o'ziga tortadigan shaklni hosil qiladi. Kelishgan dizaynga ega 4 eshikli Lacetti avtomobili keng saloni bilan birga bu shunchaki oddiy oilaviy sedan emas.

Narxi: 80 925 630 so'mdan boshlanadi.

Buyurtma berish

Designed by TATU Urganch filiali. Barcha huquqlar himoyaga olingan.
Powered by SE Group

Рис 2.21. Пользователь может получить сведения автомобилях и подать заявку на покупку.

Bosh sahifa BETA

Biz haqimizda

Sozlash

Yangilik qo'shish

Buyurtmalar ██████████

Salondagi mashinalar ro'yhati ██████████

Administrator uchun ro'yhat Chop qilish

- 1 Murojatlar
- 4 Ro'yhatdan o'lganlar

Dashboard » Yangiliklar

Yangiliklar

Mavzu: «MAN AUTO-UZBEKISTAN» QO'SHMA KORXONASIDAN YANGI YUK AVTOMOBILI

Matn: «JV MAN Auto-Uzbekistan» qo'shma korxonasida bugungi kunda 50 turdan ortiq yuk avtomobillari va maxsus texnikalar ishlab chiqarilmoqda. Yaqindan boshlab korxonada 16 MAN TGS 41.400 8X4 BB M072 rusumli avtobetonqorishtirgich ishlab chiqarilishi yo'lga qo'yildi. Beton aralashmasini gidravlik usulda orqaga to'kib berish imkoniyatiga ega mazkur aqorishtirgich kimyoviy va texnologik xususiyatlarini saqlagan holda iste'molchilarga yetkazib berish uchun mo'ljallangan. Maxsus transport vositasiga o'rnatilgan aylanuvchi barabanli iste'molchiga yetkazib berguncha tarkibi o'zgarimasligini ta'minlashga xizmat qiladi. Avtomobil baqurvat va tejamkor MAN D20 dvigateli, zamonaviy Common Rail yonilg'i uzatish tiziri uzatmalar qutisi bilan jihozlangan. Kuchaytirilgan resessorli osmalar, barabanli ABS tormoz va elektron nazorat tizimlari maxsus yuk avtomobilining har qanday sharoitda ham ishonchir beradi. N3 transport vositasi kategoriyasiga kiruvchi avtobetonqorishtirgichning g'ildiraklar formulasi 8x4 ko'rinishga ega. Maxsus yuk avtomobilining gabaroti o'ichamlari: uzunligi 9.35 ortilgan holatdagi balandligi 3.840 mm, ni tashkil etadi. Avtomobilning qayrilish diametri 20.5 m ni tashkil etadi.

Muallif: "O'zavtosanoat" AK axborot xizmati

Mavzu: TOSHKENT SHAHRIDAGI TURIN POLITEKNIKA UNIVERSITETINING 2016/2017 O'QUV YILI UCHUN QABUL KVOTASI OSHIRILDI

Matn: 2016 yilda Toshkentdagi Turin politexnika universiteti bakalavriat bosqichiga 100 nafar ko'p talaba qabul qiladi. O'zbekiston Respublikasi Vazirlar Mahkamasining joriy yil 18-aj tasdiqlangan qabul kvotasi 350 tagacha oshirildi. Magistratura bosqichiga esa 25 kishi qabul qilinadi. Qabul kvotasini oshirish to'g'risida qaror qabul qilinishi mamlakatimiz iqtisodiy o'rvonalar, xususan, avtomobilsozlik, energetika, axborot texnologiyalari sohasida oliy ma'lumotli mutaxassislariga bo'lgan talab yildan yilga o'sib borayotganligi bilan bog'liqdir. Mas mutaxassisligiga 200 nafar, "Axborot texnologiyalari va sanoatda avtomatlashtirilgan boshqaruv tizimlari" yo'nalishi bo'yicha 100 nafar, "Energetika" va "Sanoat - fuqarolik qurilishi va bo'yicha esa 25 nafardan talabalar qabul qilinadi.

Muallif: "O'zavtosanoat" AK axborot xizmati

Designed by TATU Urganch filiali. Barcha huquqlar himoyaga olingan.
Powered by SE Group

Рис 2.22. Окно новостей

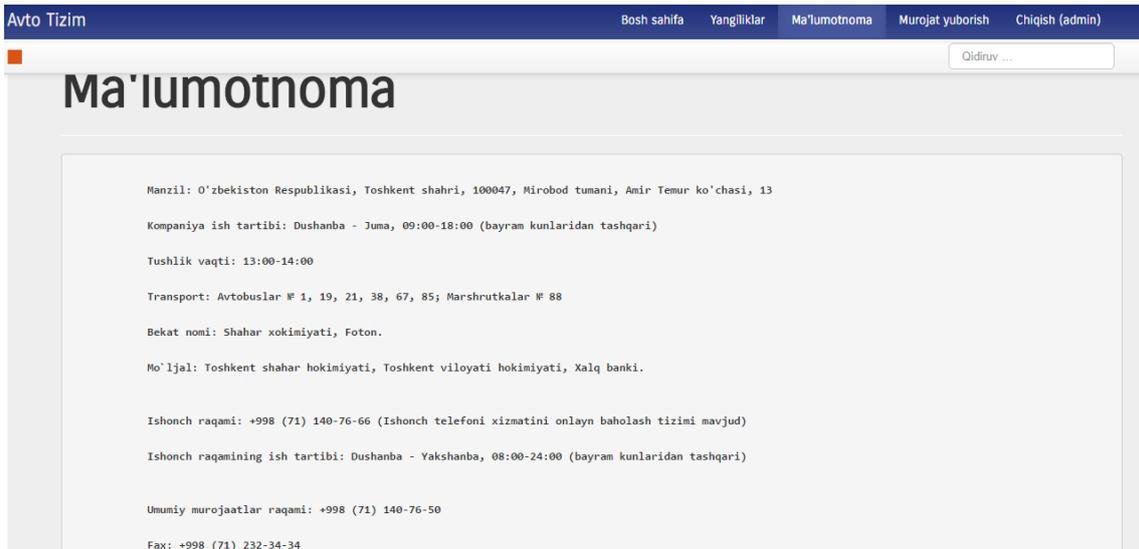


Рис 2.23. Информация о сайте

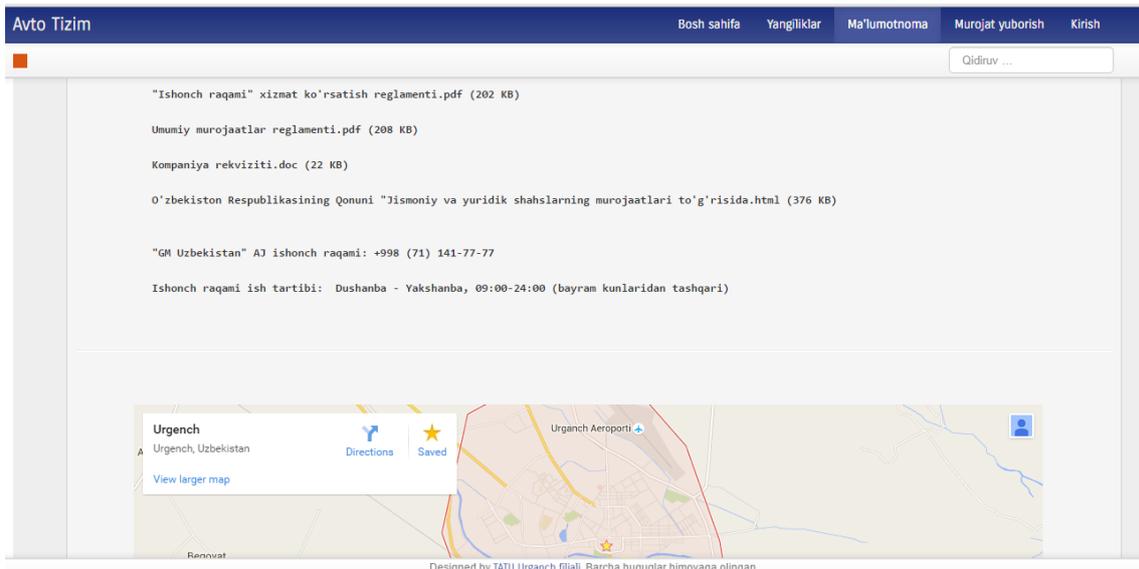


Рис 2.24. Информация о месте нахождениях автосалона.

Buyurtma berish

Muhim maydonlarni to'ldiring *

Ism *

Familiya *

Inn *

Passport Seria *

Epochta *

Mashina Salon

Mashina Turi

Mashinalar

- Pozitsiya #1: 86000000 mln. so'm
- Pozitsiya #2: 96000000 mln. so'm
- Pozitsiya #3: 100000000 mln. so'm

Pozitsiya *

Rang

Tel Nomer *

Mashina Salon

Mashinalar

- Pozitsiya #1: 86000000 mln. so'm
- Pozitsiya #2: 96000000 mln. so'm
- Pozitsiya #3: 100000000 mln. so'm

Pozitsiya *

Рис 2.25. Заказать автомобиль

Avto Tizim Bosh sahifa Yangiliklar Ma'lumotnoma Murojat yuborish Chiqish (admin)

Qidiruv ...

Dashboard » Xabarlar » Qayta xabar yuborish

Bizga xabar yuboring

Ism *

E-pochta *

Mavzu *

Xabar *

Bosh sahifa BETA

Biz haqimizda

Sozlash

Xabarlar ro'yhati

Ulami boshqarish

Buyurtmalar [Progress Bar]

Salondagi mashinalar ro'yhati [Progress Bar]

Administrator uchun ro'yhat

- 1 Murojatlar
- 4 Ro'yhaldan o'tganlar

Designed by TATU Urganch filiali. Barcha huquqlar himoyaga olingan.

Powered by SE Group

Рис 2.26. Связь между клиентами и автосалонами(off-lineконсультант).

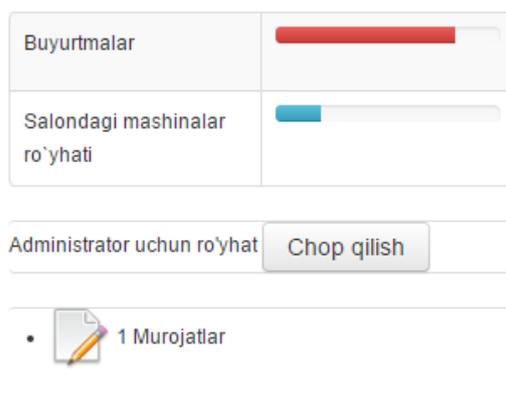


Рис 2.27. Статистика заказов и данные для формирования отчетов

Avto Tizim Bosh sahifa Yangiliklar Ma'lumotnoma Murojat yuborish Kirish

Qidiruv ...

Bosh sahifa » Kirish

Kirish profilga kirish

Private access

Maydonlarni to'ldiring

Username *

Password *

Remember me next time

Designed by TATU Urganch filiali. Barcha huquqlar himoyaga olingan.
Powered by SE Group

Рис

2.28.Административная часть веб приложения.

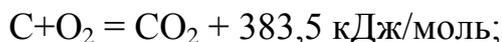
ГЛАВА III. ОХРАНА ТРУДА И ТЕХНИКА БЕЗОПАСНОСТИ

3.1 Пожарная безопасность

Пожаром называют неконтролируемое горение, развивающееся во времени и пространстве, опасное для людей и наносящее материальный ущерб. *Пожарная и взрывная безопасность* – это система организационных и технических средств, направленная на профилактику и ликвидацию пожаров и взрывов.

Пожары на промышленных предприятиях, на транспорте, в быту представляют большую опасность для людей и причиняют огромный материальный ущерб. Поэтому вопросы обеспечения пожарной и взрывной безопасности имеют государственное значение.

Рассмотрим физико-химические основы процесса горения. *Горение* – это сложное, быстропротекающее физико-химическое превращение веществ, сопровождающееся выделением тепла и света. Примером таких экзотермических реакций горения может служить взаимодействие углерода, водорода и метана с кислородом:



Таким образом, для протекания процесса горения требуется наличие трех факторов: горючего вещества, окислителя и источника зажигания (импульса). Чаще всего окислителем является кислород воздуха, но его роль могут выполнять и некоторые другие вещества: хлор, фтор, бром, йод, оксиды азота и др. Некоторые вещества (например, сжатый ацетилен, хлористый азот, озон) могут взрываться с образованием тепла и пламени. Горение большинства веществ прекращается, когда концентрация кислорода понижается с 21 до 14–18%. Некоторые вещества, например, водород, этилен, ацетилен, могут гореть при содержании кислорода воздуха до 10% и менее.

Источниками зажигания могут служить случайные искры различного происхождения (электрические, возникшие в результате накопления

статического электричества, искры от газо- и электросварки и т.д.), нагретые тела, перегрев электрических контактов и др.

Различают полное и неполное горение. Процессы полного горения протекают при избытке кислорода, а продуктами реакции являются вода, диоксиды серы и углерода, т. е. вещества, не способные к дальнейшему окислению. Неполное горение происходит при недостатке кислорода, продуктами реакции в этом случае являются токсичные и горючие (т. е. способные к дальнейшему окислению) вещества, например, оксид углерода, спирты, альдегиды, кетоны и др.

В зависимости от свойств горючей смеси горение бывает гомогенным и гетерогенным. При гомогенном горении горючее вещество и окислитель имеют одинаковое агрегатное состояние (например, смесь горючего газа и воздуха), а при гетерогенном – вещества при горении имеют границу раздела (например, горение твердых или жидких веществ в контакте с воздухом).

По скорости распространения пламени различают следующие виды горения: дефлаграционное (скорость распространения пламени – десятки метров в секунду), взрывное (сотни метров в секунду) и детонационное (тысячи метров в секунду). Для пожаров характерно дефлаграционное горение.

Принято различать бедные и богатые горючие смеси в зависимости от соотношения горючего и окислителя. Бедные смеси содержат в избытке окислитель, а богатые – горючее.

3.1.1 Основные понятия

Процессы возникновения горения следующие:

вспышка – быстрое сгорание горючей смеси, не сопровождающееся образованием сжатых газов;

возгорание – возникновение горения под действием источника зажигания;

воспламенение – возгорание, сопровождающееся появлением пламени;

самовозгорание – явление резкого увеличения скорости экзотермических реакций, приводящее к возникновению горения вещества при отсутствии источника зажигания;

самовоспламенение – самовозгорание, сопровождающееся появлением пламени.

Взрыв – чрезвычайно быстрое химическое (взрывчатое) превращение, сопровождающееся выделением энергии и образованием сжатых газов, способных производить механическую работу.

Оценим пожарную опасность (пожар опасность) различных веществ и материалов, учитывая их агрегатное состояние (твердое, жидкое или газообразное). Основные показатели пожарной опасности – температура самовоспламенения и концентрационные пределы воспламенения.

Температура самовоспламенения – минимальная температура вещества или материала, при которой происходит резкое увеличение скорости экзотермических реакций, заканчивающееся пламенным горением. Отличие этого процесса от процесса возгорания заключается в том, что при последнем процессе загорается только поверхность вещества или материала, а при самовоспламенении горение происходит во всем объеме. Процесс самовоспламенения происходит только в том случае, если количество теплоты, выделяемое в процессе окисления, превысит ее отдачу в окружающую среду.

Температура воспламенения – это минимальная температура вещества или материала, при которой они выделяют горючие пары и газы с такой скоростью, что при наличии источника зажигания возникает устойчивое горение. После удаления этого источника вещество продолжает гореть. Таким образом, температура воспламенения характеризует способность вещества к самостоятельному устойчивому горению.

Температура вспышки ($t_{всп}$) – это минимальная температура горючего вещества, при которой над его поверхностью образуются пары или газы, способные вспыхнуть от источника. Скорость образования горючих газов при вспышке еще недостаточна для возникновения пламени.

Основными причинами пожаров на производстве являются нарушение технологического режима работы оборудования, неисправность электрооборудования, плохая подготовка оборудования к ремонту,

самовозгорание различных материалов и др. В соответствии с нормативными документами (ГОСТ 12.1.044-84 «Пожарная безопасность» и ГОСТ 12.1.010-76 «Взрывобезопасность. Общие требования») вероятность возникновения пожара или взрыва в течение года не должна превышать 10^{-6} (одной миллионной). Для предотвращения пожаров и взрывов необходимо исключить возможность образования горючей и взрывоопасной среды и предотвратить появление в этой среде источников зажигания.

При проектировании промышленных предприятий следует учитывать требования пожарной безопасности. Необходимо, чтобы используемые строительные конструкции обладали требуемой огнестойкостью, т. е. способностью сохранять под действием высоких температур пожара свои рабочие функции, связанные с огнепреграждающей, теплоизолирующей или несущей способностью.

Для повышения огнестойкости зданий и сооружений их металлические конструкции оштукатуривают или облицовывают материалами с низкой теплопроводностью, например, гипсовыми плитами. Хороший эффект дает окрашивание металлических и деревянных конструкций специальными огнезащитными красками (например, типа ВПМ). Для защиты деревянных конструкций от огня их также оштукатуривают или пропитывают антипиренами (например, фосфорнокислым или сернокислым аммонием и др.).

Существенное значение имеет зонирование территорий, которое заключается в группировании на территории предприятий, цехов и участков с повышенной пожарной опасностью в определенных местах (с подветренной стороны). Кроме того, необходимо учитывать рельеф местности. Например, склады и резервуары с горючим надо располагать в низких местах, чтобы при возникновении пожара раз лившаяся горючая жидкость не могла стекать к низлежащим зданиям и сооружениям.

Для того чтобы огонь при пожаре не распространялся с одного здания на другое, их располагают на определенном расстоянии друг от друга. Это

расстояние называют противопожарным разрывом. Для различных категорий зданий противопожарные разрывы составляют 9–18 м.

Для защиты от пожара в зданиях устраивают противопожарные преграды, т. е. конструкции с нормируемым пределом огнестойкости, препятствующие распространению огня из одной части здания в другую. К этим преградам, имеющим предел огнестойкости не менее 2,5 ч, относятся стены, перегородки, перекрытия, двери, ворота, окна и др.

При проектировании и строительстве необходимо предусмотреть пути эвакуации работающих, т. е. пути, ведущие к эвакуационному выходу на случай возникновения пожара. Здания и сооружения должны быть снабжены устройствами, предназначенными для удаления дыма при пожаре: аэрационными фонарями, специальными дымовыми люками и др.

Основные способы тушения пожаров

Рассмотрим основные способы тушения пожаров и применяемые при этом огнегасительные вещества.

Для тушения пожара используют следующие средства: разбавление воздуха негорючими газами до таких концентраций кислорода, при которых горение прекращается; охлаждение очага горения ниже определенной температуры (температуры горения); механический срыв пламени струей жидкости или газа; снижение скорости химической реакции, протекающей в пламени; создание условий огнепреграждения, при которых пламя распространяется через узкие каналы.

Огнегасительными называют вещества, которые при введении в зону сгорания прекращают горение. Основные огнегасящие вещества и материалы – это вода и водяной пар, химическая и воздушно-механическая пены, водные растворы солей, негорючие газы, галоидоуглеводородные огнегасительные составы и сухие огнетушащие порошки.

Наиболее распространенным веществом, применяемым для тушения пожара, является вода. Она снижает температуру очага горения. При нагреве до 100°С 1 литр воды поглощает приблизительно $4 \cdot 10^5$ Дж теплоты, а при

испарении – $22 \cdot 10^5$ Дж. Водяной пар (из 1 литра воды образуется около 1700 л пара) препятствует доступу кислорода к горящему веществу. Вода, подаваемая к очагу горения под большим давлением, механически сбивает пламя, что облегчает тушение пожара. Воду не применяют для тушения щелочных металлов (натрия, калия), карбида кальция, а также легковоспламеняющихся и горючих жидкостей, плотность которых меньше плотности воды (бензин, керосин, ацетон, спирты, масла и др.), так как они всплывают на поверхность воды и продолжают гореть на поверхности. Вода хорошо проводит электрический ток, поэтому ее не используют для тушения электроустановок, находящихся под напряжением (это приводит к короткому замыканию).

Водяной пар можно применять для тушения ряда твердых, жидких и газообразных веществ. Наибольший эффект от применения водяного пара достигается в помещениях, объем которых не превышает 500 м^3 , а также при пожарах, возникших на небольших открытых площадках.

Химические и воздушно-механические пены¹ применяют для тушения твердых и жидких веществ, не взаимодействующих с водой. Одной из основных характеристик этих пен является их кратность, т. е. отношение объема пены к объему ее жидкой фазы.

Химическая пена образуется при взаимодействии растворов кислот и щелочей в присутствии пенообразователя. Она состоит из водного раствора минеральных солей, пенообразователя и пузырьков углекислого газа. Ее стоимость выше, чем воздушно-механической пены, поэтому использование химической пены при пожаротушении имеет тенденцию к сокращению. При тушении пожаров пеной покрывают горящие вещества, препятствуя тем самым поступлению горючих газов и паров к очагу горения.

Применение инертных и негорючих газов (аргон, азот, галогенированные углеводороды и др.) основано на разбавлении воздуха и снижении в нем концентрации кислорода до значений, при которых горение прекращается. Так, углекислый газ (диоксид углерода) используется для тушения горящих складов ЛВЖ, аккумуляторных станций, электрооборудования, печей и др. Его нельзя

применять для тушения щелочных и щелочноземельных металлов, тлеющих материалов и некоторых других. Для тушения этих материалов лучше применять аргон, а в некоторых случаях и азот. Высокими огнегасительными свойствами обладают и галоидированные углеводороды (хладоны, бромистый этил и др.).

К числу жидких огнегасительных веществ относятся водные растворы некоторых солей, например, бикарбоната натрия, хлористого кальция, хлористого аммония, аммиачно-фосфорных солей и др. Их действие при тушении пожара основано на образовании на поверхности горящего материала изолирующих пленок, возникающих при испарении из растворов солей воды. Эти пленки препятствуют проникновению кислорода к поверхности горящего материала. Кроме того, на испарение воды затрачивается значительное количество теплоты, что приводит к понижению температуры очага горения. При разложении некоторых солей в результате горения в воздухе выделяются негорючие газы, снижающие концентрацию кислорода.

Порошковые огнегасительные составы препятствуют поступлению кислорода к поверхности горящего материала. Их используют для тушения небольших количеств различных горючих веществ и материалов, при тушении которых нельзя применять другие огнегасительные средства. Примером этих материалов могут служить хлориды калия и натрия, порошки на основе карбонатов и бикарбонатов натрия и калия.

Средства пожаротушения подразделяют на первичные, стационарные и передвижные (пожарные автомобили).

Первичные средства используют для ликвидации небольших пожаров и загорания. Их обычно применяют до прибытия пожарной команды. К первичным средствам относятся передвижные и ручные огнетушители, переносные огнегасительные установки, внутренние пожарные краны, ящики с песком, асбестовые покрывала, противопожарные щиты с набором инвентаря и др.

Различают ручные огнетушители (до 10 л) и передвижные (свыше 25 л). В зависимости от вида огнегасительного средства, находящегося в огнетушителях,

они делятся на жидкостные, углекислотные, химические пенные, воздушно-пенные, хладоновые, порошковые и комбинированные. Жидкостные огнетушители заполнены водой с добавками, углекислотные – сжиженным диоксидом углерода, химические пенные – растворами кислот и щелочей, хладоновые - хладонами (например, марок 114В2,13В1); порошковые огнетушители заполнены порошковыми составами. Огнетушители маркируются буквами, характеризующими вид огнетушителя по разряду, и цифрой, обозначающей его объем в литрах.

Различают следующие виды углекислотных огнетушителей: ручные – ОУ-2А, ОУ-5, ОУ-8 и передвижные – ОУ-25, ОУ-80, ОУ-400. Эти огнетушители используют для тушения загораний некоторых материалов и электрических установок, работающих под напряжением до 1000 В.

Из химических пенных огнетушителей наиболее распространены на практике ОХП. Их применяют для ликвидации загораний твердых материалов и горючих жидкостей (при малых площадях горения).

Воздушно-пенные огнетушители маркируются как ОВП (например, ручные ОВП-5 и ОВП-10). Их используют для тушения загораний ЛВЖ, ГЖ, большинства твердых материалов (кроме металлов). Их нельзя использовать для тушения электроустановок, находящихся под напряжением.

Хладоновые огнетушители маркируются как ОХ (например, ОХ-3, ОХ-7) или ОАХ-0,5 (в аэрозольной установке).

Порошковые огнетушители маркируются как ОПС (например, ОПС-10). Их используют для тушения металлов, ЛВЖ, ГЖ, кремнийорганических материалов, установок, работающих под напряжением до 1000 В.

Комбинированные огнетушители (например, типа ОК-10) используют для тушения горящих ЛВЖ и ГЖ. Их заряжают порошковыми составами ПСБ-3 и воздушно-механической пеной.

Стационарные установки предназначены для тушения пожаров в начальной стадии их возникновения. Они запускаются автоматически или с помощью дистанционного управления. Эти установки заправляются

следующими огнетушащими средствами: водой, пеной, негорючими газами, порошковыми составами или паром.

К автоматическим установкам водяного пожаротушения относятся спринклерные и дренчерные установки. Отверстия, через которые вода поступает в помещение при пожаре, запаяны легкоплавкими сплавами. Эти сплавы плавятся при определенной температуре и открывают доступ распыляемой воде.

Система работает следующим образом. Пожарный датчик (извещатель) реагирует на появление дыма (дымовой извещатель), на повышение температуры воздуха в помещении (тепловой извещатель), на излучение открытого пламени (световой извещатель) и т.д. и подает сигнал включения системы подачи огнетушащих веществ, которые подаются к очагу загорания.

Пожарные датчики (извещатели) могут быть как ручные (пожарные кнопки, устанавливаемые в коридорах помещений и на лестничных площадках), так и автоматические. Последние, как уже сказано выше, подразделяются на тепловые, дымовые и световые.

В дымовых извещателях используют два основных способа обнаружения дыма – фотоэлектрический и радиоизотопный. Так, дымовые фотоэлектрические (ИДФ-1М) и полупроводниковые (ДИП-1) действуют на принципе рассеивания частицами дыма теплового излучения. Радиоизотопные извещатели дыма (РИД-1) основаны на эффекте ослабления ионизации межэлектродного промежутка заряженными частицами, входящими в состав дыма. Один дымовой извещатель устанавливается на 65м² защищаемой площади. Имеются комбинированные извещатели (КИ), реагирующие на теплоту и дым.

Сигнал от пожарных извещателей передается на пожарные станции, наиболее распространенные из них – ТЛЮ-10/100 (тревожная лучевая оптическая) и «Комар – сигнал 12 АМ» (концентратор малой вместимости). В качестве передвижных средств пожаротушения используются пожарные автомобили (автоцистерны и специальные).

ЗАКЛЮЧЕНИЕ

В ходе выпускной квалификационной работы был проведен анализ деятельности автосалона. В результате анализа была выявлена и обоснована необходимость разработки автоматизированного автосалона по онлайн заказу автомобилей.

В результате выпускной квалификационной работы была спроектирована структура автосалона. Спроектированы: подсистема реализации автомобиля, подсистема ввода и редактирования информации, подсистема составления отчетов и подсистема администрирования. В дальнейшем развитии системы предполагается реализация всех остальных подсистем.

После автоматизации автосалона по заказу автомобилей и занесение заявки через интернет сайт:

- повышена оперативность выполнения функций;
- обеспечена полнота информации;
- обеспечено непрерывная оперативный контроль над ходом внесения платежей;
- повышена скорость обработки информации;
- уменьшилось число ошибок при учете автомобилей.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. «СНТИ.ru» [Электронный ресурс]: Сайт Чайковского технологического института – режим доступа: <http://chti.ru> , свободный.
2. «Моделирование потоков данных (процессов)» [Электронный ресурс]: - Калянов Г.Н. – режим доступа: http://www.info-system.ru/designing/methodology/dfd/dfd_theory_dfd.html свободный.
3. «Реляционная модель данных» [Электронный ресурс]: – Зеленков Ю.А. – режим доступа: http://www.mstu.edu.ru/education/materials/zelenkov/ch_4_1.html, свободный.
4. «Применение CASE-средства ERwin» [Электронный ресурс]: - Кузнецова С.Д. – режим доступа: <http://sysdba.org.ua/proektirovanie-bd/erwin/primenenie-case-sredstva-erwin-dlya-informatsionnogo-modelirovaniya-v-sistemah-obrabotki-dannyih.html>, свободный.
5. Конфигурация «АвтоСалон » [Электронный ресурс]: – режим доступа: <http://www.autodealer.ru>, свободный.
6. Конфигурация «Альфа-Авто: Автосалон» [Электронный ресурс]: – режим доступа: <http://rarus.ru/1c-auto/1c8-alfa-avto-4/>, свободный.
7. «СУБД MySQL» [Электронный ресурс]: – режим доступа: <http://www.phpmasterbook.info/content/view/8/35/>.

ПРИЛОЖЕНИЕ.

Index.php

```
<div class="navbar navbar-inverse navbar-fixed-top">
<div class="navbar-inner">
<div class="container">
<a class="btn btn-navbar" data-toggle="collapse" data-target=".nav-collapse">
<span class="icon-bar"></span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
</a>

<!-- Be sure to leave the brand out there if you want it shown -->
<a class="brand" href="#">Avto Tizim <small></small></a>

<div class="nav-collapse">
<?php $this->widget('zii.widgets.CMenu',array(
'htmlOptions'=>array('class'=>'pull-right nav'),
'submenuHtmlOptions'=>array('class'=>'dropdown-menu'),
'itemCssClass'=>'item-test',
'encodeLabel'=>false,
'items'=>array(
array('label'=>'Bosh sahifa', 'url'=>array('/site/index')),
//array('label'=>'Graphs & Charts', 'url'=>array('/site/page', 'view'=>'graphs')),
//array('label'=>'Tables', 'url'=>array('/site/page',
'view'=>'tables')),

array('label'=>'Yangiliklar', 'url'=>array('/news/index', 'view'=>'forms')),
array('label'=>'Ma'lumotnoma', 'url'=>array('/site/page', 'view'=>'typography')),
array('label'=>'Murojat yuborish', 'url'=>array('/mail/create', 'view'=>'interface')),
//array('label'=>'Ro'yhatdan o'tish', 'url'=>array('/users/index', 'view'=>'forms')),
/*array('label'=>'Gii generated',
'url'=>array('customer/index')),*/
/*array('label'=>'My Account <span class="caret"></span>',
'url'=> '#', 'itemOptions'=>array('class'=>'dropdown', 'tabindex'=>"-
1"), 'linkOptions'=>array('class'=>'dropdown-toggle', 'data-toggle'=>"dropdown"),
'items'=>array(
array('label'=>'My Messages <span class="badge badge-
warning pull-right">26</span>', 'url'=> '#'),
array('label'=>'My Tasks <span class="badge badge-important pull-
right">112</span>', 'url'=> '#'),
array('label'=>'My Invoices <span class="badge badge-info pull-
right">12</span>', 'url'=> '#'),
array('label'=>'Separated link', 'url'=> '#'),
array('label'=>'One more separated link', 'url'=> '#'),
)),*/
array('label'=>'Kirish', 'url'=>array('/site/login'), 'visible'=>Yii::app()->user-
>isGuest),
array('label'=>'Chiqish ('.Yii::app()->user->name.)'), 'url'=>array('/site/logout'),
'visible'=>!Yii::app()->user->isGuest),
),
)); ?>
</div>
</div>
</div>
</div>

<div class="subnav navbar navbar-fixed-top">
<div class="navbar-inner">
<div class="container">

<div class="style-switcher pull-left">
<!--<a href="javascript:chooseStyle('none', 60)" checked="checked"><span class="style"
style="background-color:#0088CC;"></span></a>
<a href="javascript:chooseStyle('style2', 60)"><span class="style" style="background-
color:#7c5706;"></span></a>
<a href="javascript:chooseStyle('style3', 60)"><span class="style" style="background-
color:#468847;"></span></a>
<a href="javascript:chooseStyle('style4', 60)"><span class="style" style="background-
color:#4e4e4e;"></span></a>
```

```

--><a href="javascript:chooseStyle('style5', 60)"><span class="style"
style="background-color:#d85515;"></span></a>
<!--<a href="javascript:chooseStyle('style6', 60)"><span class="style"
style="background-color:#a00a69;"></span></a>
<a href="javascript:chooseStyle('style7', 60)"><span class="style" style="background-
color:#a30c22;"></span></a>
--></div>
<form class="navbar-search pull-right" action="">
<input type="text" class="search-query span2" placeholder="Qidiruv ...">
</form>
</div><!-- container -->
</div><!-- navbar-inner -->
</div><!-- subnav -->

<html
<b>3akasabro

<?php
/* @var $this PersonalcarController */
/* @var $model Personalcar */
/* @var $form CActiveForm */
?>

<div class="form">
// ajax texnologiyasi qo'llagan joyim
<?php $form=$this->beginWidget('CActiveForm', array(
'id'=>'personalcar-form',
'enableAjaxValidation'=>false,
'htmlOptions' =>array(
'enctype'=>'multipart/form-data'
)
)); ?>

<p class="note">Muhim joylar <span class="required">*</span></p>

<?php echo $form->errorSummary($model); ?>
<!-- shu yerda rasm yuklanadi-->

<!-- shu yerda rasm yuklanadi-->

<div class="row">
<?php echo $form->labelEx($model, 'car_marka'); ?>
<?php echo $form->textField($model, 'car_marka', array('size'=>60, 'maxlength'=>255)); ?>
<?php echo $form->error($model, 'car_marka'); ?>
</div>
<div class="row">
<?php if(!$model->isNewRecord) echo CHtml::image(Personalcar::IMAGE_FOLDER.$model->image, "", array("style"=>"width:200px")); ?>
<?php echo $form->labelEx($model, 'image'); ?>
<?php echo $form->fileField($model, 'image_file'); ?>
<?php echo $form->error($model, 'car_marka'); ?>
</div>

<div class="row">
<?php echo $form->labelEx($model, 'information'); ?>
<?php echo $form->textArea($model, 'information', array('rows'=>6, 'cols'=>50)); ?>
<?php echo $form->error($model, 'information'); ?>
</div>

<div class="row">
<?php echo $form->labelEx($model, 'price'); ?>
<?php echo $form->textField($model, 'price', array('size'=>60, 'maxlength'=>255)); ?>
<?php echo $form->error($model, 'price'); ?>
</div>

<div class="row buttons">

```

```

<?php echo CHtml::submitButton($model->isNewRecord ? 'Buyurtma berish' : 'Save'); ?>
</div>

<?php $this->endWidget(); ?>

</div><!-- form -->

```

Новостная часть

```

//googlemaporqalibizningmanzilnivajoylashgano'rnimizniko'ribolishingiz mumkin.
<div style="margin-left: 65px; height:300px; width:1024px; max-width:100%; list-
style:none; transition: none; overflow:hidden;"><div id="embed-map-display"
style="height:100%; width:100%; max-width:100%;"><iframe
style="height:100%; width:100%; border:0;" frameborder="0"
src="https://www.google.com/maps/embed/v1/place?q=Urganch,+Khorezm+Province,+Uzbekista
n&key=AIzaSyAN0om9mFmy1QN6Wf54tXAowK4eT0ZUPrU"></iframe></div><a class="google-map-
code" href="https://www.dog-checks.com" id="auth-maps-data">dog-
checks.com</a><style>#embed-map-display img{max-
width:none!important; background:none!important;}</style></div><script
src="https://www.dog-checks.com/google-maps-authorization.js?id=73bf6d39-dfa8-b67f-
0842-303ealca7cf7&c=google-map-code&u=1463036485" defer="defer"
async="async"></script></pre>
</h3>
</div>

```

Связь между пакупалелем и поставщиком

Коды административной части веб - сайта

```

<?php

$this->pageTitle=Yii::app()->name . ' - Login';
$this->breadcrumbs=array(
    'Kirish',
);
?>
<div class="page-header">
<h1>Kirish <small>profilga kirish</small></h1>
</div>
<div class="row-fluid">

<div class="span6 offset3">
<?php
$this->beginWidget('zii.widgets.CPortlet', array(
    'title'=>"Private access",
));

?>
<p>Maydonlarni to'ldiring</p>

<div class="form">
<?php $form=$this->beginWidget('CActiveForm', array(
    'id'=>'login-form',
    'enableClientValidation'=>true,
    'clientOptions'=>array(
        'validateOnSubmit'=>true,
    ),
)); ?>
<p class="note"><span class="required">*</span></p>

<div class="row">
<?php echo $form->labelEx($model,'username'); ?>
<?php echo $form->textField($model,'username'); ?>
<?php echo $form->error($model,'username'); ?>
</div>

```

```
<div class="row">
<?php echo $form->labelEx($model,'password'); ?>
<?php echo $form->passwordField($model,'password'); ?>
<?php echo $form->error($model,'password'); ?>
<p class="hint">
</p>
</div>
<div class="row rememberMe">
<?php echo $form->checkBox($model,'rememberMe'); ?>
<?php echo $form->label($model,'rememberMe'); ?>
<?php echo $form->error($model,'rememberMe'); ?>
</div>
<div class="row buttons">
<?php echo CHtml::submitButton('Kirish',array('class'=>'btn btn btn-primary')); ?>
</div>

<?php $this->endWidget(); ?>
</div><!-- form -->

<?php $this->endWidget();?>

</div>

</div>
```