

Федеральное архивное агентство

**Рекомендации по созданию
Интернет-каталогов архивных документов**

Исполнители:

зам. нач. отд. автоматизированных
архивных технологий РГАНТД

Глищинская Н.В.

вед. специалист отд. автоматизированных
архивных технологий РГАНТД

Караваев И.В.

Москва

2012

Рекомендации по созданию Интернет-каталогов архивных документов. /
И.В. Караваев, Н.В. Глищинская. Москва, 2012. 29 с., илл.

Ключевые слова: интернет-каталог, электронный каталог, база данных, архивный проект

В рекомендациях описаны виды электронных каталогов, структура каталогов, уровни и структура описания (состав полей) информации, возможность представления цифровой копии архивного документа с привязкой к описанию, приводятся практические примеры.

© Федеральное архивное агентство, 2012

© Авторы, 2012

Оглавление

Введение	4
1. Интернет каталог – разновидность веб-приложения	4
2. Базы данных и системы управления базами данных	6
3. Разновидности баз данных	7
4. Реляционные базы данных – основа Интернет-каталогов.....	8
5. MySQL – «посредник» между базой данных и пользователем.....	8
6. Преобразование базы данных из формата MS Access в формат MySQL.....	9
7. Преобразование текстового файла формата MS Word в таблицу MS Excel.....	16
8. Выбор элементов управления для создания запросов к БД	18
9. Принцип организации онлайн-доступа к БД	23
10. Оператор SELECT – основной инструмент для получения данных из БД.....	23
11. Практический пример – форма для ввода запроса и ее обработчик.....	25
12. Некоторые аспекты оптимизации производительности онлайн-каталогов.....	28
Источники	29

Введение

Размещение электронных ресурсов в сети Интернет значительно расширяют аудиторию потребителей архивной информации. Зачастую пользователи не знают, где хранятся документы по интересующей их теме. Оказать помощь в поиске необходимой информации могут размещенные на архивных сайтах путеводители, каталоги, тематические подборки, аннотированные описи, базы данных, Интернет-выставки и прочие вспомогательные материалы. Перечисленные электронные ресурсы могут не только предоставить информацию о содержании фондов архива, но и обеспечить доступ непосредственно к самим архивным документам.

Электронный каталог – информационная система, содержащая сведения об архивных документах, которые могут включать поисковые данные документа, текстовую описательную информацию, образы документов, сведения о систематизации и другое. Электронный каталог, являющийся составной частью информационно-справочной системы архива или нескольких архивов, обеспечивает оперативность получения информации широкому кругу пользователей сети Интернет.

Виды электронных каталогов:

- тематический справочник или указатель, например, Межархивный указатель «Документы по истории Отечественной войны 1812 г.», размещенный на портале «Архивы России» (<http://1812.rusarchives.ru/>);

- справочник или указатель по видам документов архива, например, Видеокаталог РГАКФД (<http://rgakfd.ru/catalog/video/>), Кинокаталог РГАНТД (<http://rgantd.ru/ecfilm/catalog.htm>);

- медиаресурс – электронный каталог с образами документов, например: Тематический каталог фото- и видеодокументов «Победа. 1941–1945», размещенный на портале «Архивы России» (<http://victory.rusarchives.ru/>), Электронный фотокаталог РГАКФД (<http://rgakfd.altsoft.spb.ru/start.do>), Коллекция фотодокументов РГАНТД (<http://rgantd.ru/elcatalog/photocat.htm>).

1. Интернет каталог – разновидность веб-приложения

Не будет преувеличением сказать, что бурное развитие сети Интернет оказало огромное влияние на многие сферы человеческой деятельности. И, конечно, Интернет способствовал поистине революционным изменениям в области разработки и применения программного обеспечения. Появилась новая категория приложений, специально

разработанных для сети Интернет и учитывающих особенности серверов Web, т.н. веб-приложения. Примеры таких приложений хорошо известны каждому пользователю Сети – это многочисленные Интернет-магазины, поисковые и справочные системы, сервисы для Интернет-телефонии и обмена сообщениями в режиме реального времени, системы хранения и передачи потокового видео через Интернет и т.д.

Для веб-приложений характерно взаимодействие с пользователем посредством Интернет-браузеров, таких как Internet Explorer, Opera, Mozilla, Chrome и др. При этом доступ к веб-приложениям осуществляется так же, как и к обычным статическим страницам HTML – по URL-адресу.

Однако по принципам своей работы веб-приложения отличаются от статических страниц HTML, т.к. при помощи веб-сервера они способны обращаться к различным активным объектам, сервисам и системам, таким, например, как базы данных (БД). Так, в ответ на запрос, введенный пользователем в окне браузера, веб-сервер способен «на лету» сформировать отчет, содержащий запрашиваемую информацию, и отобразить его на экране пользователя. При этом для получения данных веб-сервер генерирует специальный запрос к базе данных.

Важно отметить, что веб-приложения способны работать не только в Интернете, но и в локальных сетях организаций. Использование браузера на компьютере конечного пользователя в качестве основного средства доступа к базам данных значительно уменьшает затраты на сопровождение крупных локальных сетей. При этом не только упрощается процедура установки программного обеспечения на рабочие станции локальной сети, но и облегчается сопровождение баз данных и других электронных систем организации, работающих централизованно на специально выделенных серверах.

Архивные Интернет-каталоги тоже можно представить как одну из разновидностей веб-приложений, а это значит, что для их разработки и эксплуатации прекрасно подойдут технические и программные средства, применяемые для других веб-приложений.

Итак, «сердцем» любого Интернет-каталога, в том числе, архивного, является база данных, а в качестве средства доступа к информации, содержащейся в базе данных, обычно используется специальное веб-приложение, взаимодействующее с пользователем посредством любого популярного веб-браузера. С точки зрения пользователя, такое веб-приложение выглядит как обычный веб-сайт, содержащий разнообразные элементы управления (см. раздел № 8 «Выбор элементов управления для создания запросов к БД») для формирования запросов к БД.

2. Базы данных и системы управления базами данных

Появление баз данных стало самым важным достижением в области программного обеспечения. Базы данных лежат в основе информационных систем, и это коренным образом изменило характер работы многих организаций. Что же такое – база данных?

База данных – это совместно используемый набор логически связанных данных (и описание этих данных), предназначенный для удовлетворения информационных потребностей организации; это единое, большое хранилище данных, которое однократно определяется, а затем одновременно используется многими пользователями – представителями разных подразделений.

Программное обеспечение, с помощью которого пользователи могут определять, создавать и поддерживать базу данных, а также осуществлять к ней контролируемый доступ, называется системой управления базами данных (СУБД). Основная задача СУБД – предоставление пользователю базы данных возможности работы с ней, не вникая в детали работы всей системы на уровне аппаратного обеспечения.

Почему используются системы с базами данных? Какие преимущества получает пользователь, работая с такими системами? Отметим лишь некоторые из них:

- Компактность. Появляется возможность минимизировать или исключить полностью создание и ведение объемных картотек на бумажных носителях.
- Скорость работы. Выборка, сортировка и обновление данных с помощью компьютерной системы происходит гораздо быстрее, чем аналогичные действия, производимые человеком вручную.
- Снижение трудозатрат. Отпадает необходимость длительной и однообразной работы над картотекой вручную.
- Актуальность информации. За счет автоматизированных методов работы с информацией, содержащейся в БД, пользователю в любое время доступны самые свежие данные.
- Удобство управления данными, за счет их нахождения в единой централизованной БД.

Организация онлайн-доступа к БД посредством сети Интернет дает еще одно крайне важное преимущество – расширение круга потенциальных пользователей БД до практически неограниченного количества за счет снятия географических и временных ограничений.

Каким же образом можно превратить локальную архивную базу данных в Интернет-каталог, работающий в режиме доступа 365/24? Эту задачу можно решить, по крайней мере, двумя способами:

- преобразованием существующей локальной архивной БД (например, в формате MS Access или MS Excel) в формат, пригодный для функционирования в рамках веб-сервера, как правило, это MySQL;
- формированием новой архивной БД «с нуля», непосредственно в сети Интернет, с помощью созданного для этой цели веб-интерфейса.

В рамках настоящих Рекомендаций мы будем рассматривать первый из упомянутых способов, т.е. преобразование локальной БД в Интернет-каталог.

3. Разновидности баз данных

Существуют следующие разновидности баз данных:

1) Древовидные (реестр Windows, файловая система FAT, язык разметки XML и др.) – здесь информация хранится в древовидной структуре, и доступ осуществляется через «путь», т.е. последовательное указание всех узлов от корневого до нужного. Например: «C:\My_docs\My_pictures\Me.jpg». Недостатками этого способа хранения данных являются медленный поиск, если не известен путь, и низкая устойчивость к повреждениям структуры. Преимущество – возможность хранить в классифицированном виде разнородную информацию и осуществлять быстрый поиск при наличии ключа.

2) Сетевые базы данных (например, сеть Интернет) – представляющие собой совокупность узлов, обособленных друг от друга и содержащих определённую информацию. Каждый узел содержит некоторое количество ссылок на другие узлы, по которым и ведётся поиск. Недостатки – очень сложный и долгий поиск, возможно получение неполной информации или даже невозможность ее поиска вообще. Преимущества – легкое добавление любой разнородной информации, самая высокая стабильность структуры из всех систем.

3) Реляционные базы данных (от англ. relation – связь). Для простоты можно считать, что реляционная база данных – это набор двумерных простых таблиц. Недостатки реляционных баз данных – хранение только однородной информации, сложности при добавлении новых структур и взаимоотношений; информация, хранящаяся в такой БД, должна быть в нужной степени абстрагирована. Преимущества – прежде всего, очень высокая скорость поиска – по этому параметру у реляционных баз данных нет конкурентов, высокая стабильность работы, широкий выбор программного

обеспечения для их разработки и поддержки, удобство использования для различных практических задач.

4. Реляционные базы данных – основа Интернет-каталогов

Итак, реляционную базу данных можно упрощённо представить как набор взаимосвязанных таблиц. Таблица – это основной структурный элемент базы данных. Структура таблицы состоит из столбцов и строк, которые принято называть, соответственно, «полями» и «записями». Таблицы в реляционной системе являются логическими, а не физическими структурами. Это означает, что таблицы реляционной БД – это просто абстрактный способ представления физически хранящейся в базе данных информации. Такой подход значительно упрощает восприятие информации конечным пользователем, от которого оказываются скрыты сугубо технические детали вроде размещения записей в БД, их последовательность, кодировки данных, индексы и т.п.

Поля в БД всегда имеют уникальные имена (по которым их можно идентифицировать) и хранят данные только какого-либо одного типа, например, числа, текстовые строки, дату, время и т.п. Создавая таблицу, необходимо указать тип каждого из полей, заранее определившись, какого рода данные будут в них храниться. В случае архивной базы данных структура таблицы может быть следующей:

Порядковый номер – целое число;

Название архива – строка (100 символов);

Сокращенное название архива – строка (10 символов);

...

Номер фонда – число или строка (до 255 символов), в зависимости от способа нумерации;

Наименование фонда – строка (> 255 символов);

Архивный номер документа – число или строка (до 255 символов), в зависимости от способа нумерации;

... и т.д.

5. MySQL – «посредник» между базой данных и пользователем

Что такое MySQL? MySQL – это наиболее популярная система управления базами данных с открытым исходным кодом. «Открытость» означает, что вы можете бесплатно устанавливать, запускать программу и модифицировать ее исходный код (который, как и

ее саму, можно загрузить из сети Интернет). Аббревиатура SQL расшифровывается как Structured Query Language (структурированный язык запросов) – язык, применяемый для взаимодействия с большинством существующих баз данных.

MySQL – это реляционная система управления базами данных (СУБД). С технической точки зрения MySQL – программа, управляющая файлами, которые составляют базу данных, но часто термин «база данных» применяется как к самой программе, так и к этому набору файлов. БД, как упоминалось выше, это обычная совокупность взаимосвязанных данных (текстовых, числовых, двоичных), за хранение и организацию которых отвечает СУБД. Подробную документацию по MySQL можно найти, например, на сайте разработчика этой СУБД: <http://dev.mysql.com/doc/>, правда, на английском языке. Также существует большое количество «неофициальных» сайтов о MySQL со справочной информацией на русском языке.

Программные инструменты для работы с MySQL содержатся во многих распространенных языках программирования, например, в PHP, Perl и Java. Именно с помощью этих инструментов обеспечивается доступ к БД из веб-приложений и веб-сайтов.

6. Преобразование базы данных из формата MS Access в формат MySQL

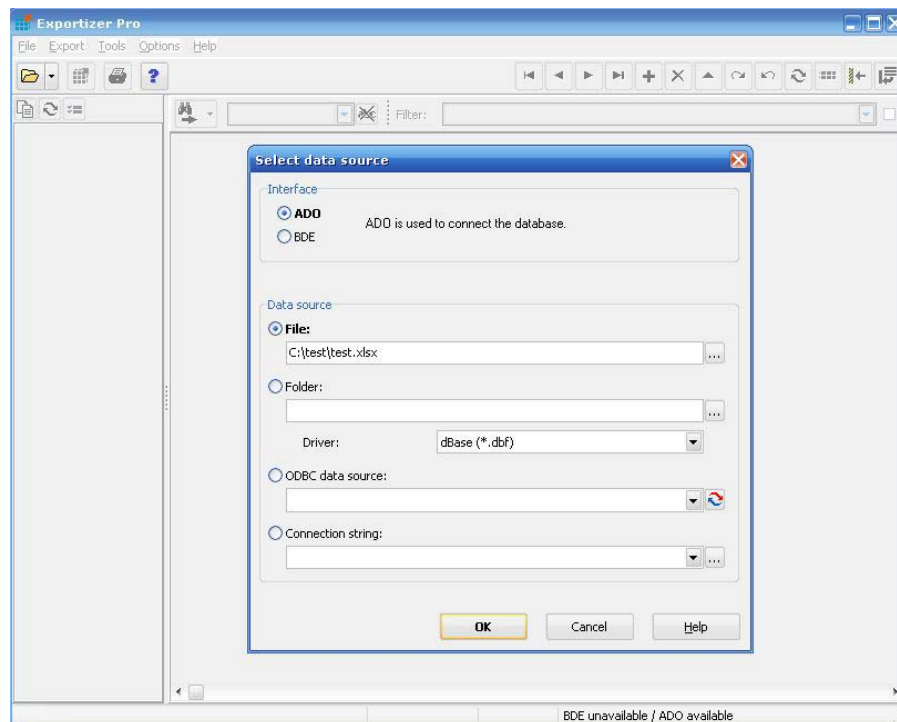
Рассматривается случай использования СУБД MS Access 2007 и БД, состоящей из 1 таблицы.

1. Откройте базу данных MS Access. Перейдите в режим просмотра таблицы.
2. На вкладке «Внешние данные» нажмите кнопку «Excel» (экспорт в таблицу MS Excel). В появившемся окне укажите имя файла для сохранения таблицы (например, test.xlsx) и формат файла (Книга Excel (*.xlsx)). Чуть ниже, в разделе параметров экспорта должна быть проставлена только одна галочка: «Экспортировать данные с макетом и форматированием». Нажмите «ОК». После того, как появится окно с сообщением, что экспорт успешно завершен, окно MS Access и, соответственно, базу данных MS Access, можно закрыть.
3. Теперь нужно преобразовать получившуюся таблицу из формата MS Excel 2007 (в нашем случае, test.xlsx) в обычный текстовый файл (*.txt), содержащий специальную разметку.

Для этого хорошо подойдет условно-бесплатная программа Exportizer Pro.

Скачайте (<http://www.vlsoftware.net/exportizer-pro/download.html>) и установите ее.

Откройте Exportizer Pro, далее из меню откройте диалог «Select data source». В этом диалоговом окне укажите имя файла, который нужно открыть (больше в этом окне ничего менять не нужно) и нажмите «ОК».



После этого в окне программы вы увидите загруженную таблицу с данными из БД.

Теперь данные из этой таблицы нужно преобразовать в текстовый формат. Для этого выполните команду меню «Export > Export to File or Database» (или просто нажмите кнопку «Export Data» на панели инструментов). Вы увидите диалоговое окно с большим количеством вкладок и настроек. Выберите вкладку «Text», в поле «File» укажите имя будущего текстового файла (например, test1812.txt), а также выполните все остальные настройки в соответствии с рисунком, расположенным ниже, и нажмите кнопку «Next»:

Schema: Standard;

Field separator: Other («|» – вертикальная черта);

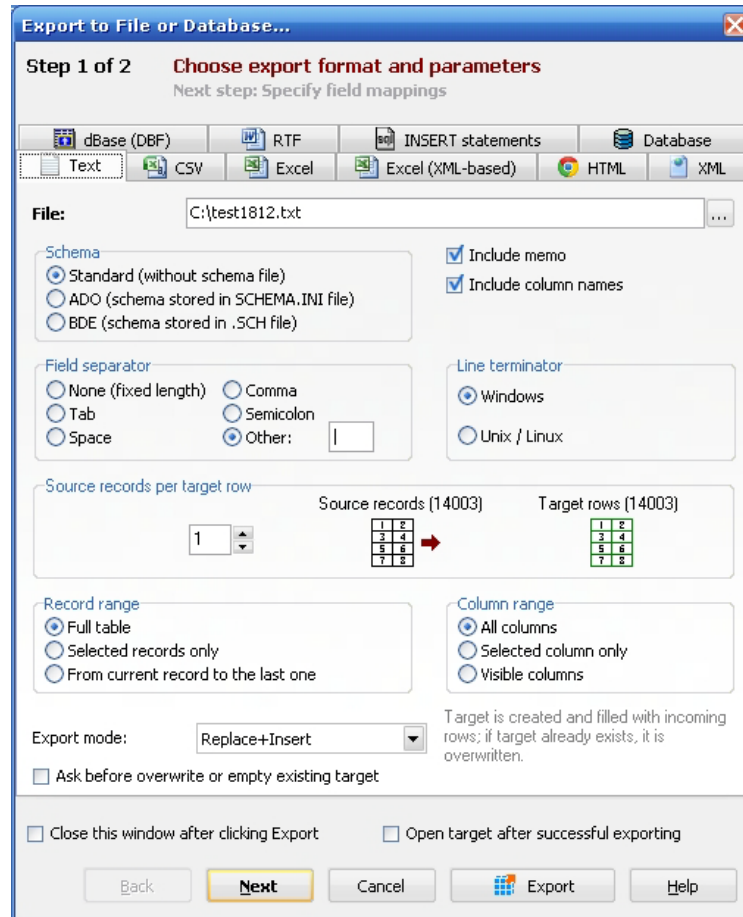
Line terminator: Windows;

Source records per target row: 1;

Record range: Full table;

Column range: All columns;

Export mode: Replace+Insert (означает, что если текстовый файл с таким именем уже существует, то он будет перезаписан).



После нажатия кнопки «Next» вы увидите еще одно диалоговое окно с заголовком «Step 2 of 2. Specify field mappings». В этом окне можно ничего не менять, а можно, например, поменять названия полей таблицы и их взаимное расположение, при необходимости. В нашем случае ничего менять не нужно. Просто нажмите кнопку «Export». Вы увидите быстро увеличивающийся счетчик обработанных надписей, а после этого окно с сообщением типа «10550 records have been exported to file C:\test1812.txt». Это означает, что преобразование завершено успешно.

Теперь окно программы Exportizer Pro можно закрыть.

Итак, мы преобразовали информацию из таблицы MS Access в текстовый файл, причем столбцы исходной таблицы в каждой строке этого файла отделены друг от друга вертикальными чертами (можете убедиться в этом, открыв test1812.txt в любом текстовом редакторе, например, в «Блокноте»).

4. Создайте на сервере базу данных и таблицу внутри этой базы, например, с помощью графической серверной оболочки phpMyAdmin. Обратите внимание, что вы должны создать таблицу с тем же количеством столбцов, что и у исходной таблицы MS Access! Каждому столбцу вновь созданной таблицы присвойте те же имена, что и у

столбцов таблицы MS Access, а также корректный тип данных (например, для полей, в которых будут числа – INT, для полей с текстовыми строками – TEXT).

Замечание: названия столбцам таблицы необходимо давать на английском языке, а не на русском.

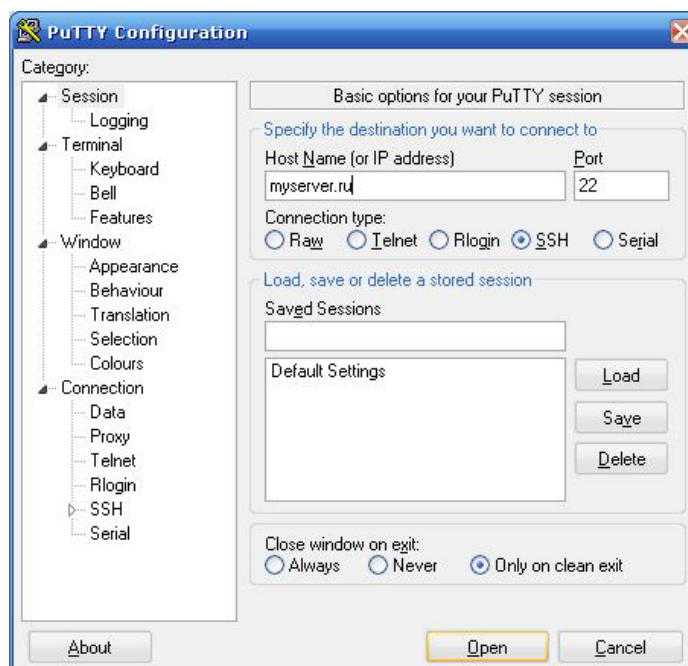
Также помните, что имя создаваемой таблицы и имя созданного ранее текстового файла должны совпадать.

5. Теперь нужно экспортировать данные из получившегося текстового файла в таблицу MySQL, созданную на сервере.

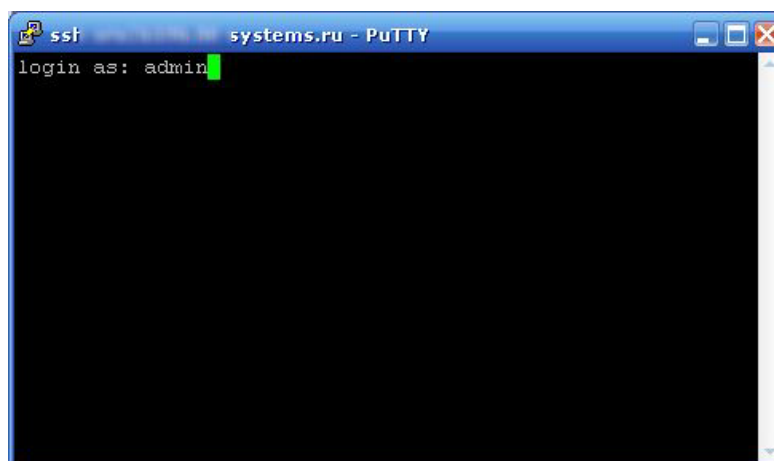
Это можно сделать через SSH соединение. SSH (Secure SHell) – это программа для входа и выполнения команд или программ на удаленных компьютерах, а также для передачи файлов с одного компьютера на другой. SSH обеспечивает строгую проверку подлинности и безопасности соединения по незащищенному каналу связи и поэтому очень удобна для удаленного администрирования веб-серверов.

Подключиться к серверу по протоколу SSH можно с помощью любого SSH-клиента. Например, бесплатной программой PuTTY (вы можете скачать ее по адресу: <http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe> или на любом другом сайте).

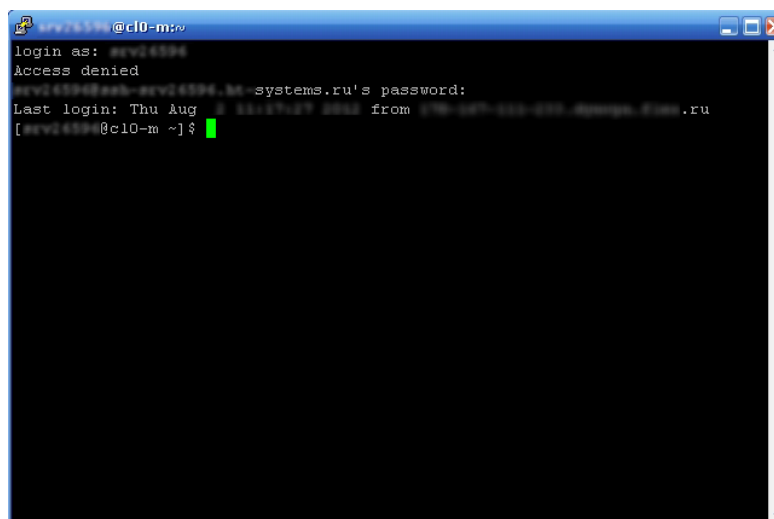
Запустив Putty, вы увидите окно «PuTTY Configuration», в котором нужно указать лишь адрес сервера, к которому вы хотите подключиться. Напишите имя сервера в поле «Host Name (or IP address)» и нажмите «Open». Откроется окно, в котором нужно ввести сначала имя пользователя (login as), а затем и пароль (password). Если доступ с такими учетными данными разрешен, вы увидите запрос командной строки SSH.



Окно ввода имени сервера для подключения



Окно ввода имени пользователя и пароля.



Запрос командной строки SSH.

6. Скопируйте текстовый файл, полученный в пункте №3 на сервер, это можно сделать с помощью любой программы, поддерживающей доступ к серверу по протоколу FTP (от англ. File Transfer Protocol – протокол передачи файлов), например, FreeCommander.

Теперь в командной строке PuTTY перейдите в каталог, куда вы только что скопировали текстовый файл, и выполните следующую команду (будьте внимательны, каждый символ имеет значение!):

```
mysqlimport --local -u <имя пользователя БД> -p --fields-terminated-by="|" --lines-terminated-by="\r\n" <имя БД (а не таблицы!)> <имя текстового файла.txt>
```

Замечание: вводить эту команду нужно именно в командной строке SSH, а не в режиме монитора SQL – там она просто не выполнится, будет появляться сообщение об ошибке.

После ввода этой команды потребуется ввести пароль для доступа к базе данных, соответствующий указанному в команде имени пользователя.

Замечание: имя текстового файла (до расширения .txt) должно совпадать с именем таблицы, куда предполагается экспортировать данные. Например, если имя текстового файла test1812.txt, то имя таблицы в БД должно быть test1812.

Если вам требуется помощь по синтаксису команды `mysqlimport`, то нужно ввести следующую команду:

```
mysqlimport --help
```

Если ошибок при экспорте не было, вы увидите надпись вида: «Records: 10550 Deleted: 0 Skipped: 0 Warnings: 0»

7. Проверьте содержимое созданной на сервере таблицы с помощью phpMyAdmin – вы должны увидеть там всю информацию из исходной таблицы MS Access.

Возможная проблема:

В отдельных случаях вы можете заметить, что текст в некоторых ячейках новой, преобразованной в формат MySQL, базы данных «обрезан», часто прямо посередине слова, хотя в исходном файле *.xlsx в этой же ячейке текст присутствует полностью, без сокращений. Как нетрудно убедиться, длина такого «обрезанного» текста составляет всего 255 символов. В случае преобразования базы данных, содержащей архивные сведения, такая ситуация, безусловно, неприемлема, так как БД может содержать, в том числе, поля с описаниями архивных документов, длина которых может значительно превышать 255 символов.

В чем же причина этой проблемы и как ее разрешить?

Оказывается, проблема возникает еще на этапе преобразования БД из формата MS Excel в текстовый файл.

По умолчанию, драйвер, отвечающий за преобразование файла, сначала считывает первые 8 строк исходной таблицы и, если не обнаруживает полей, содержащих более 255 символов текста, считает, что все поля исходной таблицы также содержат не более 255 символов. Даже если в 9 записи таблицы будет поле длиной более 255 символов, оно будет усечено до 255 символов.

Решение вытекает из самой проблемы. Если исходная таблица содержит поля с текстовыми данными, длина которых более 255 символов, необходимо перенести одну из

записей с «длинным» полем в самое начало таблицы. Например, если первое «длинное» поле встречается в 25 записи, переместите эту запись, сделав ее, например, второй или третьей от начала таблицы.

Если в таблице есть несколько «длинных» полей, например, описание документа и URL документа, необходимо переместить в начало таблицы каждую из строк, содержащих «длинные» поля.

Пример:

Ниже – исходная таблица, «длинные» поля есть в строках № 16, 19, 20. Если преобразовать таблицу в таком виде, эти поля будут обрезаны до 255 символов.

Номер записи	Поле 1	Поле 2	Поле 3	Поле 4	Поле 5
1					
2					
3					
* * *					
16			Текст > 255 символов		
17					
18					
19				Текст > 255 символов	
20	Текст > 255 символов				

Поэтому необходимо изменить порядок строк, например, так:

Номер записи	Поле 1	Поле 2	Поле 3	Поле 4	Поле 5
1					
16			Текст > 255 символов		
19				Текст > 255 символов	
20	Текст > 255 символов				
2					
3					
4					
5					

Как вы можете видеть, теперь записи с «длинными» полями находятся на 2, 3 и 4 позиции от начала таблицы. При преобразовании длина соответствующих полей сохранится, потому что полям № 1, № 3 и № 4 будет автоматически присвоен «длинный» формат.

Можно поступить проще. Создайте в начале таблицы «искусственную» строку, в которой каждое поле, обязанное содержать «длинные» данные, заполните текстом с заведомо большей, чем 255 символов, длиной. После преобразования таблицы эту строку можно будет удалить (ниже эта строка отмечена курсивом).

Номер записи	Поле 1	Поле 2	Поле 3	Поле 4	Поле 5
0	<i>Текст > 255 символов</i>		<i>Текст > 255 символов</i>	<i>Текст > 255 символов</i>	
1					
2					

Ссылка на описание этой проблемы и другие методы ее решения можно найти на официальном сайте компании Microsoft: <http://support.microsoft.com/kb/281517>.

7. Преобразование текстового файла формата MS Word в таблицу MS Excel

Возможно, исходным материалом для будущего Интернет-каталога будет обычный текстовый файл в формате MS Word. Для того чтобы воспользоваться способом, описанным в предыдущем разделе, сначала нам нужно преобразовать документ MS Word в таблицу MS Excel, а затем – в текстовый файл, подходящий для конвертации в таблицу MySQL.

Исходные данные:

Файл MS Word с определенным количеством наборов строк текста, разделенных символом абзаца. В свою очередь, каждый из наборов строк отделен от последующего двумя символами абзаца, например:

Набор № 1

Душа моя озарена неземной радостью, как эти чудесные весенние утра.

Я так счастлив, мой друг, так упоен ощущением покоя, что искусство мое страдает от этого.

Ни одного штриха не мог бы я сделать, а никогда не был таким большим художником, как в эти минуты.

Набор № 2

Как бы выразить, как бы вдохнуть в рисунок то, что так полно.

Друг мой... Но нет! Мне не под силу это, меня подавляет величие этих явлений.

Душа моя озарена неземной радостью, как эти чудесные весенние утра.

Набор № 3

Я совсем один и блаженствую в здешнем краю, словно созданном для таких как я.

Я так счастлив, мой друг, так упоен ощущением покоя, что искусство мое страдает от этого.

Ни одного штриха не мог бы я сделать, а никогда не был таким большим художником, как в эти минуты.

Предполагаемый результат:

Все строки в каждом из наборов нужно объединить в одну – это будут «строки» будущей таблицы MS Excel.

Затем скопировать все «объединенные» строки в Excel и разбить по соответствующим столбцам.

Как это сделать?

Безусловно, если объем данных незначителен, можно создать таблицу в MS Excel и вручную, а затем скопировать строки из MS Word в соответствующие столбцы таблицы. Но что делать, если будущая таблица должна содержать несколько сотен, или даже тысяч строк? В этом случае процесс преобразования целесообразнее максимально автоматизировать, выделив весь текст и используя команду «Заменить» в MS Word. Итак:

1. Необходимо избавиться от лишних пробелов в исходном тексте (замените два пробела на один, продолжайте делать эту замену до того момента, когда очередная замена приведет к нулевому результату, т.е. двойных пробелов в тексте не останется).

2. Замените два идущих подряд символа абзаца на один символ абзаца плюс, например, символ вертикальной черты (|). В результате ваши наборы строк будут отделены друг от друга символом, отличным от простого абзацного переноса, это пригодится в дальнейшем.

3. Замените символ абзаца на символ, например, решетки (#).

4. Выделите весь текст и преобразуйте его в таблицу, состоящую из 1 столбца, указав в качестве разделителя символ вертикальной черты (для этого выполните команду Вставка > Таблица > Преобразовать в таблицу).
5. Получившуюся таблицу вставьте на новый лист MS Excel.
6. Теперь осталось лишь разделить строки в MS Excel с помощью команды «Данные > Текст по столбцам», указав в качестве разделителя символ решетки (#).
7. Проверьте и исправьте мелкие недочеты и несоответствия в получившейся таблице.

8. Выбор элементов управления для создания запросов к БД

Для формирования запросов к онлайн-БД используются HTML-формы. Веб-страница может содержать любое количество форм, но одновременно на сервер может быть отправлена только одна из них. По этой причине данные форм должны быть независимы друг от друга. Формы могут состоять из различных элементов, имеющих свой уникальный внешний вид и функциональное назначение. Количество элементов формы неограниченно. Рассмотрим некоторые из этих элементов.

Наиболее известными и часто используемыми элементами форм являются **текстовые поля**. В них посетитель веб-страницы может вводить любую информацию. Текстовые поля ввода создаются с помощью тега <INPUT>, в котором атрибуту «type» присваивается значение «text»:

```
<INPUT type="text" name="text1" size="n" maxlength="m">
```

где type - тип поля, name – уникальное имя поля как элемента формы, size - размер видимой части текстового поля на экране, maxlength – максимальное количество символов, которое пользователь может ввести в это поле.

Пример HTML-кода формы, содержащей текстовое поле ввода, представлен ниже:

```
<html>
<head>
<title>HTML-формы</title>
</head>
<body>
<p>Введите ваш запрос</p>
<form method="get" action="script.php">
<input name="text1" type="text" size="30" maxlength="50" />
<input name="button1" type="submit" value="Отправить">
</form>
</body>
</html>
```

В браузере эта форма будет выглядеть так:

Введите ваш запрос

При загрузке веб-страницы на экране появится текстовое поле, в которое пользователь должен будет ввести свой запрос и нажать кнопку «Отправить». После этого данные из формы будут переданы файлу-обработчику, указанному в атрибуте «action» тега «form». В данном случае, файлу script.php.

Флажки являются еще одним элементом форм. Они применяются в ситуациях, когда пользователь должен выбрать один, несколько или ни одного из параметров поиска, путем установки или снятия «галочки». Флажки создаются с помощью тега <INPUT>, в котором атрибуту «type» присваивается значение «checkbox»:

```
<INPUT name="checkbox" type="checkbox" checked>
```

где type - тип поля, name - имя поля как элемента формы, checked – атрибут, который используется для того, чтобы отметить флажок, как выделенный, если требуется, например, задать некий параметр «по умолчанию».

Пример HTML-кода формы, содержащей флажки, представлен ниже:

```
<html>
<head>
<title>HTML-формы</title>
</head>
<body>
<p>Выберите один или несколько вариантов</p>
<form method="get" action="script.php">
<p>
<label>
<input type="checkbox" name="CheckboxGroup1" value="Вариант 1" checked>Вариант
1</label><br>
<label>
<input type="checkbox" name="CheckboxGroup1" value="Вариант 2">Вариант 2</label><br>
<label><input type="checkbox" name="CheckboxGroup1" value="Вариант 3" checked>Вариант
3</label><br>
</p>
<input name="button1" type="submit" value="Отправить">
</form>
</body>
</html>
```

В браузере эта форма будет выглядеть так:

Выберите один или несколько вариантов

Вариант 1

Вариант 2

Вариант 3

При загрузке веб-страницы на экране появятся группа флажков, один или несколько из которых пользователь должен отметить и нажать кнопку «Отправить». После этого данные из формы будут переданы файлу-обработчику, указанному в атрибуте «action» тега «form». В данном случае, файлу script.php.

Обратите внимание, что в случае отсутствия в коде атрибута «value», из формы на сервер будет передана лишь информация о том, был ли выбран данный флажок или нет. Если флажок был установлен пользователем, то форма передаст обработчику значение «on» (т.е., флажок установлен). В противном случае, форма не передаст никаких данных.

Если вам требуется передать на сервер «содержимое» флажка (т.е. текст, который помещен рядом с ним), необходимо указать этот текст в атрибуте «value» тега <INPUT>, как это показано в коде выше.

При использовании нескольких флажков каждый из них является отдельным элементом. Как указывалось выше, пользователь может выбрать один, несколько флажков или не выбрать ни одного. При этом если пользователем будут отмечены все флажки, обработчик получит содержимое их всех. Поэтому каждый флажок должен иметь собственное уникальное значение.

Переключатели в HTML-формах используются в тех случаях, когда имеется набор возможных параметров, но выбрать можно только один из них. В отличие от флажков, для связи переключателей друг с другом необходимо задать всем переключателям одной группы одинаковое имя.

Переключатели создаются с помощью тега <INPUT>, в котором атрибуту «type» присваивается значение «radio»:

```
<INPUT name="radio" type="radio" checked>
```

где type - тип поля, name - имя поля как элемента формы, checked - атрибут, который используется для того, чтобы отметить переключатель как выделенный, если требуется, например, задать некий параметр «по умолчанию».

Пример HTML-кода формы, содержащей переключатели, представлен ниже:

```

<html>
<head>
<title>HTML-формы</title>
</head>
<body>
<p>Выберите один из вариантов</p>
<form method="get" action="script.php">
<p>
<label>
<input type="radio" name="RadioGroup1" value="Вариант 1">Вариант 1</label><br>
<label>
<input type="radio" name="RadioGroup1" value="Вариант 2" checked>Вариант 2</label><br>
<label>
<input type="radio" name="RadioGroup1" value="Вариант 3">Вариант 3</label><br>
</p>
<input name="button1" type="submit" value="Отправить">
</form>
</body>
</html>

```

В браузере эта форма будет выглядеть так:

The screenshot shows a rectangular form box with a black border. At the top, the text "Выберите один из вариантов" is centered. Below it, there are three radio buttons, each followed by its label: "Вариант 1", "Вариант 2", and "Вариант 3". The radio button for "Вариант 2" is selected, indicated by a small black dot in the center of the circle. At the bottom of the form, there is a rectangular button with rounded corners and a light gray background, containing the text "Отправить".

При загрузке веб-страницы на экране появляется группа переключателей, из которых пользователю необходимо отметить только один и нажать кнопку «Отправить». После этого содержимое атрибута «value» тега INPUT отмеченного переключателя будет передано файлу-обработчику, указанному в атрибуте «action» тега «form». В данном случае, файлу script.php.

Списки или **выпадающие списки** представляют собой элементы форм, включающие в себя, как правило, несколько объектов (пунктов списка). Списки создаются с помощью тегов <SELECT> и <OPTION>. Тег <SELECT>, отвечающий за непосредственное создание списка, содержит в себе несколько тегов <OPTION>, каждый из которых содержит текст, соответствующий одному элементу списка:

```

<SELECT name="select" multiple size="m">
<OPTION value="value 1">
<OPTION value="value 2">
...
<OPTION value="value n">
</SELECT>

```

где name - имя элемента формы, multiple – атрибут для указания возможности множественного выбора элементов списка, size – размер видимой части списка, value – значение переменной, которое будет передано файлу-обработчику.

Пример HTML-кода формы, содержащей обычный и выпадающий списки, представлен ниже:

```
<html>
<head>
<title>HTML-формы</title>
</head>
<body>
<p>Выберите один из элементов списка</p>
<form method="get" action="script.php">
<select name="List1" size="1">
<option value="Вариант 1">Вариант 1</option>
<option value="Вариант 2">Вариант 2</option>
<option value="Вариант 3">Вариант 3</option>
</select>
<p>Выберите один или несколько элементов списка</p>
<select name="List2[]" multiple>
<option value="Вариант 1">Вариант 1</option>
<option value="Вариант 2">Вариант 2</option>
<option value="Вариант 3">Вариант 3</option>
</select>
<br>
<input name="button1" type="submit" value="Отправить">
</form>
</body>
</html>
```

В браузере эта форма будет выглядеть так:

При загрузке веб-страницы на экране появляются два списка, в первом из которых необходимо выбрать один из пунктов, а во втором – один или несколько пунктов, и нажать кнопку «Отправить». После этого содержимое атрибутов «value» тегов OPTION обоих списков будет передано файлу-обработчику, указанному в атрибуте «action» тега «form». В данном случае, файлу script.php.

9. Принцип организации онлайн-доступа к БД

Удобство размещения информации в базе данных становится особенно заметным при организации многопользовательского доступа к ней через Интернет.

Для работы с MySQL используются различные языки, в том числе Perl, Java, C, C++ и Python, но чаще других – язык PHP. Одной из причин этого является открытость исходных кодов PHP и его тесная интеграция с MySQL.

Для создания Интернет-каталога вам потребуются:

1. База данных, размещенная на сервере.
2. Наличие у вас прав доступа к этой базе на уровне администратора, а также специальной «пользовательской» учетной записи с ограниченными правами доступа, с помощью которой и будет происходить взаимодействие БД и веб-интерфейса. Использование пользовательской учетной записи необходимо, прежде всего, для защиты БД от несанкционированного доступа извне.

3. Интерфейс для доступа к БД.

Интерфейс можно условно разделить на визуальный, видимый пользователю и содержащий упоминавшиеся ранее элементы форм (текстовые поля, флажки, переключатели и т.д.), и программный, непосредственно отвечающий за «логику» работы Интернет-каталога и недоступный для просмотра пользователем.

10. Оператор SELECT – основной инструмент для получения данных из БД

Оператор SELECT является основой любого запроса, обращенного пользователем к БД, и предназначается для извлечения информации из таблиц БД. Структура этого оператора на первый взгляд достаточно проста, однако на практике, при выполнении сложных многоуровневых запросов, требуется ее «расширение» и уточнение. Оператор SELECT состоит из нескольких отдельных ключевых слов, или *условий*, некоторые из которых являются обязательными, а другие – нет.

Упрощенно схему построения оператора SELECT можно представить в виде:

SELECT <имя_столбца(ов)> **FROM** <имя_таблицы (таблиц)> **WHERE** <условия_поиска> **ORDER BY** <имя_столбца(ов)> **LIMIT** <m, n>, где:

SELECT – это основное условие, его наличие обязательно. Оно используется для определения столбцов, которые вы хотите получить в результате выполнения своего запроса. Можно явно перечислить имена столбцов через запятую, а можно заменить их

символом звездочки *, что означает требование выбрать все столбцы, имеющиеся в таблице.

FROM – это второе по важности условие оператора SELECT, оно также является обязательным и используется для определения таблиц, из которых должны извлекаться столбцы, перечисленные в условии SELECT. Столбцы можно извлекать как из одной, так и из нескольких таблиц одновременно.

WHERE – важное, хотя и необязательное условие, используемое для фильтрации строк, возвращаемых условием FROM.

ORDER BY – необязательное условие, предназначенное для упорядочивания результатов запроса. Содержит имена столбцов, по которым осуществляется сортировка результатов запроса. Столбцы перечисляются в порядке убывания приоритета. В первую очередь записи сортируются по столбцу, указанному первым. По умолчанию сортировка производится по возрастанию (ASC). Если порядок сортировки нужно изменить, следует указать рядом с именем столбца ключевое слово DESC.

С помощью условия LIMIT можно ограничивать число записей, которые возвращаются в таблице результатов запроса. Если задан параметр <m>, то будет выведена таблица, содержащая указанное количество строк, начиная с первой. Если указаны оба числа, разделенные запятой, то параметр <m> определяет номер строки, с которой начинается выборка, а параметр <n> – размер этой выборки.

Наибольший интерес с точки зрения создания Интернет-каталога представляет именно условие WHERE, поскольку оно позволяет извлекать из БД не все строки, а лишь те, которые соответствуют определенному критерию, в частности, пользовательскому запросу. Условие WHERE содержит условия поиска, которые применяются оператором SELECT в качестве фильтра для выбора только требуемых строк из полного набора результатов. Условие поиска может содержать одно или несколько выражений сравнения, которые можно объединять с помощью логических операторов AND (и) и OR (или) для конструирования более сложных запросов. Применительно к архивным Интернет-каталогам, в условии WHERE могут использоваться:

- операторы сравнения: = (равно), <> (не равно), < (меньше), > (больше), <= (меньше или равно), >= (больше или равно).
- ключевое слово BETWEEN, позволяющее проверять попадание искомого результат в указанный диапазон значений. Диапазон определяется с помощью двух типизированных выражений, разделенных ключевым словом AND (и).

– ключевое слово LIKE служит для проверки совпадения строки символов с указанным образцом. В частности, может использоваться для поиска в БД введенного пользователем текста. Образец символьной строки может состоять из любой комбинации обычных символьных строк, а также двух специальных групповых символов: процента (%) и подчеркивания (_). Символ процента может заменять один, несколько или ни одного произвольного символа, а символ подчеркивания – только один произвольный символ.

Например:

'%ая' – строка символов может быть любой длины, но должна заканчиваться на «ая» (красная, длительная, плохая и т.п.);

'каталог_' – строка символов может иметь длину только 8 символов, последний символ – любой (каталога, каталогу, каталоге, каталог2 и т.п.).

11. Практический пример – форма для ввода запроса и ее обработчик

Код формы (например, размещенный в файле form.html) для ввода пользовательского запроса может выглядеть так:

```
<html>
<head>
<title>Форма для ввода пользовательского запроса к БД</title>
</head>
<body>
<form method="POST" action="script.php">
<input type="hidden" name="posted" value="true">
<ol>
<li>
<label for="field">Выберите критерий поиска:</label>
<INPUT name="field" type="radio" value="field2" checked>Поле № 5<br />
<INPUT name="field" type="radio" value="field5">Поле № 8
</li>
<li>
<label for="zapros">Введите запрос:</label>
<input name="zapros" type="text" maxlength="60">
</li>
<li>
<input name="button1" type="submit" value="Отправить">
</li>
</ol>
```

```
</form>
</body>
</html>
```

В этой форме есть группа из двух переключателей, предназначенная для выбора критерия поиска (т.е., в каком столбце таблицы БД будет производиться поиск), текстовое поле для ввода запроса и кнопка для передачи введенных пользователем данных обработчику script.php, указанному в теге FORM.

Код обработчика может выглядеть так:

```
<html>
<head>
<title>Обработка пользовательского запроса и вывод данных из БД</title>
</head>
<body>
<?php // указание на то, что далее находятся команды на языке PHP
// Установка значения переменных для доступа к базе данных.
$db_user = "user"; // имя пользователя БД
$db_pass = "password"; // пароль пользователя БД
$db_host = "localhost"; // адрес сервера БД (обычно localhost)
$db_name = "database"; // имя БД
if (isset($_POST['posted']))
    {
    $field = $_POST['field'];
    $zapros = $_POST['zapros'];
    trim($zapros); // удаление пробелов из начала и конца строки пользовательского запроса
    if ($field == "" or $zapros == "") // проверка - введены ли данные в форму? Если нет - вывод
предупреждения
    { echo "<h3>Ошибка:</h3>";
    echo "<p>Отсутствуют параметры поиска.</p>";
    echo "<p>Пожалуйста, вернитесь на предыдущую страницу и <a
href='form.html'>уточните параметры поиска</a>.</p>";
    exit; } // остановка обработчика
$db = mysql_connect($db_host, $db_user, $db_pass); // установка соединения с БД
@mysql_query ("set CHARACTER SET utf8"); // установка кодировки символов utf-8 - может помочь
при решении проблем с отображением данных БД
if (!$db) // проверка - произошло ли подключение к БД, если нет - вывод предупреждения
    { echo "<h3>Ошибка:</h3>";
    echo "<p><b>Невозможно подключиться к MySQL серверу. Пожалуйста, попробуйте
позже.</b></p>";
    exit; }
    @mysql_SELECT_db($db_name); // выбор базы данных
```

```

$query = "SELECT * FROM table WHERE ".$field." LIKE ".$zapros."%' ORDER BY field1 DESC, field2
ASC, field3 DESC"; // запрос к БД (к таблице table), сортировка по полям field1 (убыв.), field2 (возр.),
field3 (убыв.).
$result = mysql_query($query); // результат выполнения запроса помещается в переменную result
для последующего вывода
$num_rows = mysql_num_rows($result); // подсчет количества результатов
echo "<h3>Результат поиска:</h3>"; // вывод результатов поиска
echo "<p>По запросу «<b>".$zapros."</b>»." найдено записей: <b>".$num_rows."</b>.</p>";
echo "<p>Вы можете вернуться на предыдущую страницу и <b><a href='\"index.shtml\"'>изменить
параметры поиска</a></b>.</p>";
if ($num_rows <> 0) { // если результат поиска не нулевой, то выводим данные
echo "<table>\n"; // Создание таблицы с результатами
echo "<tr> // Создание шапки таблицы
<th width='\"7%\"'>Поле № 1</th>
<th width='\"32%\"'>Поле № 2</th>
<th width='\"16%\"'>Поле № 3</th>
<th width='\"9%\"'>Поле № 4</th>
<th width='\"20%\"'>Поле № 5</th>
<th width='\"6%\"'>Поле № 6</th>
<th width='\"10%\"'>Поле № 7</th>
</tr>\n";
for ($i=0; $i<$num_rows; $i++) // вывод данных удобно организовать в цикле
{ $row = mysql_fetch_array($result); // обработка массива результатов запроса
echo "<tr>
<td>$row[field1]</td>
<td>$row[field2]</td>
<td>$row[field3]</td>
<td>$row[field4]</td>
<td>$row[field5]</td>
<td>$row[field6]</td>
<td>$row[field7]</td>
</tr>\n";
}
echo "</table>\n"; }
}
?>
</body>
</html>

```

12. Некоторые аспекты оптимизации производительности онлайн-каталогов

Не старайтесь поместить в базу данных всю информацию, которая у вас есть. Например, не нужно хранить там картинки, хотя MySQL и позволяет это делать. Помещая в базу образы графических файлов, вы замедляете работу сервера. Прочитать файл с картинкой с диска гораздо проще и, с точки зрения потребляемых ресурсов, экономичнее. Вместо картинок лучше хранить в MySQL информацию, на основе которой можно генерировать ссылки на статические картинки в динамически создаваемых скриптами документах.

Если таблицы содержат большое число столбцов или вы используете в запросах операции соединения, не следует употреблять оператор «SELECT *» лишь для того, чтобы оградить себя от необходимости набирать названия отдельных полей. Явное указание имен необходимых для работы полей таблицы БД позволяет сэкономить несколько циклов процессора при каждом запуске обработчика.

Говоря об операторе SELECT, отметим, что нужно использовать его выражение WHERE с максимальной пользой. Если в параметре WHERE содержатся наименования нескольких столбцов, производительность будет зависеть от того, в какой очередности эти столбцы записаны. На первом месте выражения WHERE должен стоять столбец, возвращающий минимальный набор записей, на втором – столбец, возвращающий следующий по «минимальности» набор записей, и так далее для всех оставшихся столбцов.

Поиск по БД идет быстрее, если вместо LIKE '%строка%', употреблять LIKE 'строка%'. Операции с шаблонами регулярных выражений кешируются только в том случае, если вначале строки отсутствует символ %. Поэтому при построении поисковых запросов с LIKE избегайте начальных символов %.

Источники

1. Грабер М. Введение в SQL. – М.: Лори, 2010 – 228 с.
2. Молиново Э. SQL. Сборник рецептов. – Пер. с англ. – СПб: Символ-Плюс, 2009. – 672 с., ил.
3. Ларри Ульман. MySQL: Пер. с англ. – М.: ДМК Пресс, 2004 – 352 с.: ил. (Quick Start) MySQL.
4. Коннолли, Томас, Бегг, Каролин. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. 3-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2003. – 1440 с.: ил. – Парал. тит. англ.
5. Майкл Дж. Хернандес, Джон Л. Вьескас. SQL-запросы для простых смертных. Практическое руководство по манипулированию данными в SQL. – М.: Лори, 2003.
6. Материалы сайта <http://www.orenport.ru>
7. Материалы сайта <http://ru.wikipedia.org>
8. Материалы сайта <http://articles.org.ru>