

Администрирование базы данных MySQL

МуSQL относится к числу баз данных, обеспечивающих наиболее простое администрирование на всех платформах. Кроме того, эта база данных не требовательна к ресурсам, поэтому может эксплуатироваться даже на персональных компьютерах с небольшим объемом оперативной памяти и жесткого диска. В связи с этим разработчики, использующие средства PHP, уже в течение продолжительного времени применяют MySQL для создания полной локальной среды разработки для Web на клиентских компьютерах всевозможных типов, даже на портативных компьютерах. Многие разработчики настолько успешно освоили эксплуатацию собственных инсталляций MySQL, что имеют возможность работать дома или в дороге, используя любую предпочитаемую ими операционную систему. Кроме того, руководители групп разработчиков часто рекомендуют своим подчиненным использовать отдельные локальные инсталляции MySQL, для того чтобы не возникали предпосылки нарушения работы всей группы разработчиков из-за отказа в единственном месте. А во многих проектах с открытым исходным кодом на основе PHP ко всем разработчикам, желающим принять участие в работе, предъявляется требование — продемонстрировать исчерпывающее владение навыками администрирования базы данных MySQL.

Любой разработчик, работающий в среде PHP, вполне способен самостоятельно администрировать базу данных MySQL, в отличие от некоторых других баз данных. Для того чтобы еще больше упростить задачу администрирования, можно воспользоваться многочисленными инструментальными средствами, которые не только предусмотрены в самой базе данных MySQL, но и могут быть получены от сторонних разработчиков. К тому же, чтобы иметь возможность инсталлировать, эксплуатировать и отлаживать Web-приложения в составе многих прикладных пакетов на основе PHP, не только коммерческих, но и с открытым исходным кодом, необхо-

ГЛАВА

14

В этой главе...

Лицензирование MySQL

Инсталляция
программного
обеспечения MySQL:
переход на версию 4

Администрирование
учетных записей MySQL

Программа PHPMyAdmin

Создание резервных
копий

Репликация

Восстановление

Резюме

димо освоить работу с базой данных MySQL. Поэтому освоение навыков администрирования MySQL принесет читателю большую пользу, даже если он не собирается самостоятельно заниматься разработкой всего необходимого ему кода PHP.

Лицензирование MySQL

Прежде чем установить какую-либо часть программного обеспечения с открытым исходным кодом, необходимо тщательно изучить все связанные с этим вопросы лицензирования. Это требование является особенно важным в отношении таких программных продуктов, как MySQL, которые имеют двойные лицензии — и на коммерческую программу, и на программу с открытым исходным кодом.

К сожалению, ко времени написания данной книги правила лицензирования MySQL пересматривались, поэтому на время возникали несовместимости с лицензией PHP. До тех пор, пока такая ситуация не разрешится окончательно, разработчики, использующие язык PHP, должны соблюдать предельную осторожность, чтобы обеспечить соблюдение всех лицензий. Такая задача еще больше усложняется для тех разработчиков, которые распространяют (а не просто используют) базу данных MySQL в составе коммерческого программного продукта или программы с открытым исходным кодом.

В некоторых из последних выпусков PHP4 в поставку PHP входили клиентские библиотеки MySQL. Летом 2003 года компания MySQL AB (создатель базы данных MySQL) решила принять лицензию GPL (General Public License — общедоступная лицензия) для некоммерческого использования. Эта схема лицензирования во многих отношениях является более простой и менее ограничительной по сравнению с применявшейся ранее, но такая лицензия оказалась несовместимой с лицензией PHP в стиле Apache, поэтому группа разработчиков PHP Group посчитала себя не вправе и дальше распространять библиотеки этой базы данных. Ко времени выхода данной книги из печати указанный вопрос был на этапе обоюдного согласования, что должно было позволить распространять бесплатно результаты “совместных работ” с открытым исходным кодом, но читатель все равно должен проверить, закрыт ли окончательно данный вопрос, чтобы не иметь никаких сомнений на этот счет. Окончательные сведения о лицензировании MySQL можно найти по адресу www.mysql.com/doc/en/Licensing_and_Support.html.

Правила лицензирования имеют отдельные разделы для коммерческого использования и для использования в соответствии с лицензией GPL, и эти два варианта применения MySQL, по-видимому, являются наиболее подходящими для большинства разработчиков, использующих язык PHP.

Прежде чем приступить к развертыванию базы данных MySQL, необходимо внимательно прочитать лицензию. Ниже описаны некоторые широко распространенные конфигурации программного обеспечения, для которых может иметь значение новая схема лицензирования MySQL.

- **Применение исключительно в составе Web-узла.** Если база данных MySQL эксплуатируется исключительно как часть коммерческого или некоммерческого Web-узла, то ее можно продолжать эксплуатировать, не задумываясь о проблемах лицензирования. В таком случае компания MySQL AB предлагает лишь заключить контракт на сопровождение, позволяющий воспользоваться результатами дальнейших разработок.
- **Проект с открытым исходным кодом, разрабатываемый в соответствии с лицензией GPL.** Если вы распространяете сервер или клиентские библиотеки MySQL в составе проекта с открытым исходным кодом, выпущенного по всем правилам в соответствии с лицензией GNU General Public License, то можете (и даже обязаны) распространять программный продукт MySQL и его исходный код бесплатно.

- **Коммерческая поставка.** Если вы распространяете программное обеспечение сервера или клиентские библиотеки MySQL в составе коммерческого программного продукта, то обязаны приобрести коммерческую лицензию в компании MySQL AB.
- **Проект с открытым исходным кодом, разрабатываемый в соответствии с лицензией, отличной от GPL.** В настоящее время такая ситуация может в принципе стать источником проблем, поскольку после ввода программного обеспечения сервера или клиентских библиотек MySQL в состав проекта с открытым исходным кодом, на который не распространяется лицензия GPL, может оказаться, что ваш код подлежит лицензированию согласно GPL. Просмотрите страницу с правилами лицензирования MySQL, чтобы получить наиболее актуальную информацию о том, какой статус должна иметь база данных MySQL по отношению к вашему проекту.

Но следует учитывать, что подобных ограничений, связанных с лицензированием, во многих случаях можно легко избежать. Для этого необходимо прекратить самому поставлять базу данных и потребовать от пользователей, чтобы они приобретали и устанавливали MySQL самостоятельно.

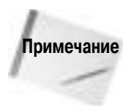
Инсталляция программного обеспечения MySQL: переход на версию 4

Следует учитывать, что сервер и клиентские библиотеки MySQL необходимо установить перед инсталляцией PHP! Безусловно, это требование не является строго обязательным во всех обстоятельствах, особенно если речь идет о системе Windows, но при использовании PHP всегда рекомендуется обеспечить инсталляцию должным образом всех серверов и библиотек сторонних разработчиков и только после этого заниматься настройкой системы PHP для обеспечения связи с этими программами.

Проблемы, которые должны быть решены до начала инсталляции

В течение продолжительного времени была доступна только версия MySQL 3, и за этот период многие разработчики PHP научились успешно ее эксплуатировать. Но в версии MySQL 4 были внесены некоторые дополнения и изменения, как со стороны базы данных, так и со стороны PHP. Поэтому и начинающие, и опытные разработчики Web должны найти время, чтобы ознакомиться с этими изменениями. Даже специалист, который имеет большой опыт работы с версией MySQL 3, не должен рассчитывать на то, что сумеет установить и эксплуатировать версию MySQL 4, используя точно такие же процедуры, как и прежде.

Перед тем как приступить к инсталляции новых версий MySQL и PHP, необходимо ознакомиться с тремя важными вопросами, относящимися к MySQL: новые и несовместимые со старыми клиентские библиотеки, новое расширение `mysqli` системы PHP и новые типы таблиц, которые обеспечивают транзакционную поддержку.



Примечание

Под транзакционной поддержкой подразумевается способность обрабатывать группу операций с базой данных как отдельную единицу работы, с точки зрения того, должны ли быть приняты или отвергнуты данные, которые вводятся или модифицируются с помощью всех этих операций. Например, транзакция в системе электронной торговли может состоять из нескольких этапов и затрагивать много разных таблиц. Предположим, что вначале регистрируется новый пользователь,

происходит получение информации об оплате и поставке, отмечается уменьшение запаса товаров на складе и т.д. Но все эти изменения не должны быть внесены в базу данных до тех пор, пока не произойдет успешное списание денежных средств с кредитной карточки, даже если каждый из предыдущих этапов процесса покупки будет выполнен успешно. В таком случае необходимо иметь транзакционную базу данных, которая позволяет отслеживать все изменения данных, происходящие в процессе оформления покупки, после чего либо фиксировать все эти изменения в виде одной единицы работы, либо выполнять откат всех этих изменений как отдельной единицы. До выпуска версии 4 база данных MySQL не была транзакционной, но теперь обеспечивает поддержку транзакций.

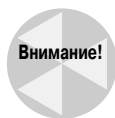
Новые клиентские библиотеки

Клиентские библиотеки для версий MySQL 3 и 4.0 не обеспечивают прямой совместимости с сервером MySQL 4.1 и последующих версий. Поэтому, если требуется эксплуатировать MySQL 4.1 и последующие версии, то необходимо повторно собрать программное обеспечение PHP с новыми библиотеками. Обратное утверждение не является истинным, поскольку более старые версии сервера MySQL можно эксплуатировать с клиентскими библиотеками MySQL 4.1. После перехода на новые библиотеки может также потребоваться обновить таблицы с правами доступа и все столбцы, содержащие хэш-массивы паролей, вычисленные системой MySQL, которые были созданы в каких-либо таблицах.

Основное различие между клиентскими библиотеками касается вопроса аутентификации. Применение функции `PASSWORD()` языка MySQL раньше приводило к получению 16-байтового хэш-массива. Начиная с 4.1.1 и последующих версий, применение этой функции приводит к получению 41-байтового хэш-массива. В системе MySQL функция `PASSWORD()` используется для установки собственных схем определения прав доступа пользователей этой системы, поэтому необходимо обновить таблицы прав доступа в версиях MySQL, предшествующих 4.1.1, с помощью сценария `mysql_fix_privilege_tables`. Не исключена также возможность, что потребуются изменить вручную в других таблицах характеристики столбцов, в которые вводятся выходные данные из функции `PASSWORD()` системы MySQL, для того чтобы они имели длину 41 байт. Фактическое содержимое этих столбцов изменять не требуется (версия MySQL 4.1 продолжает принимать хэш-массивы короче 41 байта), но размеры столбцов должны быть увеличены, для того чтобы в них можно было записывать новые значения.

Расширение `mysqli`

Буква “i” в названии расширения “`mysqli`” является сокращением от `improved` (усовершенствованный). Новое расширение `mysqli` для PHP позволяет получить доступ к новым функциональным возможностям MySQL 4.1 и последующих версий, особенно к средствам поддержки транзакций, которые относятся к числу наиболее важных новых возможностей MySQL 4.



Расширение `mysqli` может применяться только с версиями MySQL ниже 4.1, а расширение `mysqli` работает только с MySQL 4.1 и последующими версиями.

К сожалению, это расширение не является полностью совместимым со старым расширением `mysql` и теми функциями, которые связаны с `mysql`. Поэтому на этапе компиляции программного обеспечения PHP лучше всего выбрать либо то, либо другое расширение. Ко времени написания данной книги расширение `mysqli` считалось экспериментальным, поэтому, скорее всего, не должно использоваться на производстве.

Теоретически возможно откомпилировать программное обеспечение PHP одновременно с расширениями `mysql` и `mysqli` (если, например, необходимо эксплуатировать на одном и том

же компьютере какую-то версию 3.x, а также 4.1 или следующую версию MySQL), но требуется соблюдать предельную осторожность, чтобы избежать конфликтов между клиентскими библиотеками. На практике обычно лучше выбрать либо то, либо другое расширение.



Если вы решили опробовать расширение `mysql_i`, то не забудьте отменить расширение `mysql`, которое обычно разрешено по умолчанию. (В версиях Unix используйте для этого опцию `--without-mysql`, а в операционной системе Windows закомментируйте вызов расширения `mysql.dll` в файле `php.ini`.)

Таблицы с поддержкой транзакций

На протяжении почти всего периода существования программного обеспечения MySQL в нем использовались таблицы специализированного типа, который обозначается как MyISAM. На последних этапах цикла выпуска версий 3.x в программное обеспечение этой базы данных были введены два новых типа (таблица ISAM и `heap` — динамическая область памяти), но они не нашли широкого распространения. Таблицы типа MyISAM применяются по умолчанию до сих пор и намного превосходят по своей распространенности таблицы всех прочих типов.

Но для поддержки транзакций в MySQL 4.1 и последующих версиях группой разработчиков MySQL были созданы два новых типа таблиц с поддержкой транзакций — InnoDB и BDB. Если в приложении необходимо использовать операции фиксации и отката, то следует откомпилировать программное обеспечение MySQL с учетом способности распознавать один из этих типов и определять каждую таблицу как таблицу InnoDB. Кроме того, предусмотрена возможность совместно применять в одной и той же базе данных таблицы разных типов, а также преобразовывать таблицы MyISAM в таблицы InnoDB. Совместное использование таблиц разных типов может быть также основано на том принципе, чтобы таблицы InnoDB использовались в ведущей базе данных, которая поддерживает операции записи, а в ведомых базах данных по-прежнему применялись таблицы MyISAM, которые в данном случае обеспечивают поддержку только операций чтения.

Следует тщательно продумать вопрос о том, действительно ли требуются таблицы с поддержкой транзакций. Применение этих таблиц связано с возникновением значительных дополнительных издержек и поэтому приводит к замедлению; сами таблицы занимают больше места на жестком диске и требуют для своего обслуживания дополнительные инструментальные средства и процедуры. Если используются таблицы с поддержкой транзакций, то некоторые операции, такие как восстановление базы данных после сбоя, должны выполняться во многом иначе по сравнению с прежними операциями и могут оказаться сложнее (несмотря на то, что к использованию подобных операций, вероятно, придется прибегать немного реже). С другой стороны, если база данных MySQL должна эксплуатироваться на предприятии в таких условиях, когда требуются поддержка транзакций, выполнение блокировки на уровне строки, внешние ключи и горячее резервное копирование, то необходимо изучить возможности применения таблиц InnoDB.

Таблицы с поддержкой транзакций другого типа, BDB, основаны на использовании машины хранения BerkeleyDB, которая разработана компанией Sleepycat Software. Таблицы BDB не поддерживают некоторые из дополнительных средств таблиц InnoDB, таких как внешние ключи и блокировка на уровне строки, и не совсем ясно, какая компания должна предоставлять поддержку подобной установки.

Безусловно, таблицы с поддержкой транзакций все еще остаются не столь широко распространенными и, по мнению авторов, используются в основном в таких ситуациях, когда у компании, эксплуатирующей прикладное программное обеспечение, имеется достаточно ресурсов, чтобы привлечь к работе администраторов базы данных, специализирующихся в этой области, и приобрести необходимые инструментальные средства, поэтому основная часть данной главы посвящена описанию таблиц MyISAM. Дополнительную информацию о табли-

цах InnoDB можно получить по адресу www.innodb.com или www.mysql.com/doc/en/InnoDB.html.



Двоичная версия сервера MySQL для операционной системы Windows формируется с поддержкой таблиц типа InnoDB, разрешенной по умолчанию. Но фактически создаваемые таблицы не относятся к типу InnoDB, если явно не указано, что должны быть созданы таблицы этого типа.

Загрузка программного обеспечения MySQL

Все загружаемые файлы, относящиеся к программному обеспечению MySQL, находятся по адресу www.mysql.com/downloads/index.html. Выберите необходимый номер версии, а также укажите требуемую платформу настолько точно, насколько это возможно.

Одной из особенностей программного обеспечения MySQL, в отличие от большинства других серверов с открытым исходным кодом, является то, что компания, разрабатывающая эту базу данных, предпочитает поддерживать инсталляцию на основе двоичного, а не исходного кода. Безусловно, могут возникать такие ситуации, когда требуется сформировать исполняемый файл самостоятельно, но, как правило, этого следует избегать при любой возможности.

В настоящее время база данных MySQL иногда поставляется в составе дистрибутивов Linux или других пакетов, но если требуется установить более современную версию этой базы данных, то лучше деинсталлировать существующую версию с использованием инструментальных средств, предусмотренных для конкретной платформы, а затем инсталлировать новую версию.

Инсталляция MySQL в операционной системе Windows

В последнее время применяемая по умолчанию процедура инсталляции MySQL в любой версии Windows стала еще проще, поскольку теперь эта база данных поступает пользователю, аккуратно упакованная в один пакет вместе с программой инсталляции. Достаточно просто загрузить инсталляционный пакет, разархивировать его в любом удобном каталоге, а затем вызвать на выполнение программу `setup.exe`. Программа инсталляции сопровождает пользователя на протяжении всего несложного процесса и по умолчанию инсталлирует все программное обеспечение в каталоге `C:\mysql`, выбор которого для этой цели, по-видимому, является столь же подходящим, как и выбор любого другого каталога.

Сразу после инсталляции можно проверить работу сервера, запустив его из приглашения к вводу команд. Перейдите в тот каталог, где находится исполняемый файл сервера, `mysqld` (этим каталогом, по всей видимости, будет `C:\mysql\bin`), и введите команду `mysqld --console`

Если инсталляция была выполнена успешно, то на экране появятся некоторые сообщения с информацией о запуске сервера и вводе в действие поддержки InnoDB. Если же запуск сервера не происходит, то, по-видимому, возникали проблемы с правами доступа. Убедитесь в том, что каталог, в котором находятся данные, доступен для того пользователя (скорее всего, `mysql`), от имени которого эксплуатируются процессы базы данных.

Безусловно, компания MySQL AB предусмотрела новую и удобную программу инсталляции, но не воспользовалась всеми возможностями пользовательского интерфейса Windows. Предпочтительным способом эксплуатации сервера MySQL, клиентских программ и инструментальных средств все еще является выполнение команд из приглашения к вводу команд.

Программа MySQL не вводит сама себя в меню запуска **Start**, и даже не предусмотрен достаточно удобный способ останова сервера с помощью графического интерфейса пользователя. Таким образом, если вы обычно запускаете сервер, дважды щелкнув на исполняемом файле `mysqld`, то не забудьте перед останом компьютера остановить процесс сервера вручную (с помощью программы `mysqladmin`, окон **Task List**, **Task Manager** или других специализированных средств Windows).

Еще одна довольно странная особенность, которая значительно осложняет работу пользователей Windows, в отличие от пользователей Unix, состоит в том, что в настоящее время справочное руководство по языку MySQL поступает к пользователям Windows в виде одного огромного файла HTML или текстового файла. Это руководство входит в состав дистрибутива для Windows, а также предоставляется отдельно в виде архивного файла. Этот файл настолько велик, что с ним невозможно даже успешно работать, если компьютер Windows — не очень мощный и быстродействующий. По возможности вместо него загрузите с Web-узла MySQL упакованную версию в виде файла формата tar (tar-файла), в которой для каждой главы руководства предусмотрен отдельный файл HTML. Эти файлы можно извлечь на компьютере Unix, а затем скопировать на компьютер Windows. Еще один вариант состоит в том, что всегда возможно использовать документацию, предоставленную для доступа в оперативном режиме, если у вас есть надежный доступ к Интернет.

Версия PHP для операционной системы Windows поступает с поддержкой MySQL, разрешенной по умолчанию, поэтому после инсталляции этой базы данных можно сразу же приступать к работе (разумеется, после выполнения всех необходимых операций управления пользователями, которые рассматриваются в одном из последних разделов этой главы). Если требуется отключить поддержку расширения `mysql` в пользу расширения `mysqli`, то необходимо закомментировать строку `mysql` и раскомментировать строку `mysqli` в разделе `modules` файла `php.ini`.

Инсталляция MySQL в операционной системе Unix

По возможности используйте одну из двоичных версий MySQL, причем лучше всего ту из них, для которой предусмотрена программа инсталляции. На некоторых платформах (особенно на платформе Linux) приходится загружать серверное и клиентское программное обеспечение отдельно, а на других платформах предусмотрены удобные инсталляционные пакеты, содержащие все необходимое. В настоящее время имеется богатый выбор двоичных инсталляционных пакетов, поэтому для большинства пользователей не возникает необходимости собирать программное обеспечение MySQL вручную. Некоторые пакеты распространяются сторонними разработчиками, такими как компания Debian, а не компанией MySQL AB. Если не сможете найти нужную двоичную версию на узле `mysql.com`, проведите поиск в том источнике двоичных пакетов (специально предназначенных для вашей платформы), который вы обычно используете.

Точное место размещения программ и файлов данных MySQL изменяется в очень широких пределах, в зависимости от того, какой пакет используется и из какого источника он получен. В руководстве, которое находится на узле `mysql.com`, предусмотрен отдельный раздел с описанием инсталляционных компоновок, но содержащиеся в нем данные применительно к некоторым платформам часто бывают неприменимыми или неточными. Наиболее часто используемыми каталогами являются `/usr`, `/usr/local` и `/var`.

Если приходится применять универсальный двоичный инсталляционный пакет вместо удобной версии, в которой предусмотрена инсталляционная программа, то для установки базы данных MySQL может потребоваться выполнить нескольких дополнительных этапов. Введите следующие строки в приглашении к вводу команд, чтобы создать нового пользователя `mysql` и инсталлировать MySQL для эксплуатации от имени данного пользователя (для этого необходимо вначале перейти в учетную запись `root`):

```

groupadd mysql
useradd -g mysql mysql
cd /usr/local
gunzip < /path/to/mysql-VERSION-OS.tar.gz | tar xvf -
ln -s full-path-to-mysql-VERSION-OS mysql
cd mysql
scripts/mysql_install_db
chown -R root .
chown -R mysql data
chgrp -R mysql .
bin/mysqld_safe --user=mysql &

```

Пользователи версии MySQL 3.x должны учитывать, что новый сценарий запуска для MySQL теперь называется `mysqld_safe`, а не `safe_mysqld`. Тем не менее последнее имя все еще будет сохраняться как символическая ссылка на протяжении определенного переходного периода для обратной совместимости.

Теперь все готово для создания исполняемого файла PHP с клиентскими библиотеками MySQL. Используйте опцию `--with-mysql=/path/to/mysql` для более старых версий MySQL или опции `--without-mysql --with-mysqli=/path/to/mysql_config` для MySQL 4.1 и последующих версий. Обратите внимание на то, что в версии MySQL 4 необходимо задавать ссылку на фактическое местонахождение файла `mysql_config`, а не просто на каталог MySQL. Сценарий `mysql_config` — это инструментальное средство, позволяющее предоставить системе информацию об условиях компиляции клиентов MySQL, в частности, о местонахождении библиотек.

Инсталляция MySQL в операционной системе Mac OS X

В настоящее время компания MySQL AB предоставляет дистрибутив с программой инсталляции двоичного исполняемого файла, специально предназначенной для OS X. В состав этого дистрибутива входит образ диска, а не файл формата `tar`. Просто загрузите файл с расширением `.dmg` и дважды щелкните на появившейся пиктограмме. Программа инсталляции сопровождает пользователя на протяжении всего процесса и предлагает заданный по умолчанию путь инсталляции.

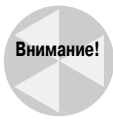
Пользователи программы Mac Internet Explorer могут обнаружить, что файл MySQL загружается под именем `download.php`, а не `mysql-standard-4.x.x.dmg`. В таком случае дождитесь завершения загрузки, а затем измените имя файла.

Вспомогательные работы, выполняемые после инсталляции

Поставка MySQL выполняется с пустым паролем для пользователя `root` базы данных MySQL. Как только будет завершена успешная инсталляция базы данных и клиентов (и по возможности еще до того, как будет создан исполняемый файл PHP с поддержкой MySQL), необходимо задать пароль пользователя `root` следующим образом:

```
mysqladmin -u root password 'new_password';
```

Очевидно, что при этом вместо приведенного выше обозначения `new_password` необходимо задать действительный пароль.



Ни при каких обстоятельствах не следует допускать даже мысли о том, чтобы в приведенной выше команде в качестве пароля пользователя `root` применялся пароль пользователя `root` серверного компьютера! Пользователь `root` сервера и пользователь `root` базы данных не имеют никакого отношения друг к другу. Кроме того, в качестве пароля пользователя `root` базы данных не следует применять свой обычный пароль пользователя. Настоятельно рекомендуем не экономить усилия и подготовить не использовавшийся ранее пароль.

Пользователям Unix может также потребоваться ввести имя каталога MySQL в состав переменной `PATH`, чтобы не нужно было каждый раз вводить полное имя пути для вызова на выполнение клиентской программы с интерфейсом командной строки. При использовании командного интерпретатора `bash` соответствующая команда будет выглядеть следующим образом:

```
export PATH=$PATH:/usr/local/mysql
```

Откорректируйте эту команду в соответствии с требованиями применяемого командного интерпретатора. Если запись с указанием данного каталога будет добавлена в строку `PATH` файла запуска командного интерпретатора (например, `.bashrc`), то не придется выполнять этот шаг после каждой регистрации на компьютере.

Теперь сервер MySQL готов к эксплуатации.

Основные клиентские команды MySQL

На первых порах у пользователей вызывает удивление информация о том, что двоичный файл `mysql`, находящийся в каталоге `mysql/bin`, представляет собой не серверную, а клиентскую программу (серверная программа имеет имя `mysqld`). После ввода имени `mysql` в приглашении командного интерпретатора происходит вызов клиентской программы MySQL с интерфейсом командной строки, позволяющей получить доступ к одному из серверов MySQL.

Чтобы подключиться к серверу MySQL с использованием клиента с интерфейсом командной строки, необходимо ввести следующую простую команду:

```
mysql [-h hostname] [-P portnumber] -u username -p
```

При этом почти наверняка потребуется указать имя пользователя `username`, так как в противном случае клиентская программа будет пытаться использовать имя пользователя командного интерпретатора. Если не будет указана опция пароля `-p`, то программа `mysql` проверит, требуется ли пароль для того пользователя, который пытается подключиться к серверу, и если пароль действительно требуется, но не задан, отвергнет попытку подключения. Если происходит подключение к локальному хосту, то опцию `-h` с именем хоста `hostname` задавать не требуется, а если подключение происходит к применяемому по умолчанию порту (3306), не нужно задавать опцию `-P` с номером порта `portnumber`. Для команды MySQL предусмотрен целый ряд других опций, но обычно на первых порах достаточно применить описанные здесь опции. Предположим, что при подключении указано имя пользователя `root`; в таком случае будет выведено приглашение к вводу пароля пользователя `root`, который был задан на предыдущем этапе.

После подключения к серверу необходимо выбрать используемую базу данных. Для этого применяется следующая команда:

```
USE dbname;
```

Точку с запятой в этой команде вводить не обязательно, но все прочие команды SQL должны оканчиваться точкой с запятой, поэтому следует привыкнуть к ее использованию. До того как будут созданы новые базы данных, во вновь выполненной инсталляции имеются только две базы данных: `mysql` и `test`. Если вы только что подключились к серверу MySQL

как пользователь `root`, то имеете доступ к обеим базам данных, а в случае подключения под именем любого другого пользователя вы получите доступ только к базе данных `test`.

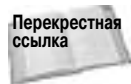
Команда `SHOW TABLES;` позволяет получить список всех таблиц в используемой базе данных.

Чтобы быстро ознакомиться со структурой одной из таблиц базы данных, можно воспользоваться командой `SHOW COLUMNS FROM tablename;` Эта команда выводит имена всех столбцов с указанием типов, размеров, применяемых по умолчанию значений, и другую полезную информацию.

Для просмотра всех значений в таблице достаточно ввести оператор `SELECT` без ограничительных условий:

```
SELECT * FROM tablename;
```

Однако при использовании подобной команды необходимо соблюдать осторожность, поскольку в действующих базах данных результаты выполнения запроса такого рода могут оказаться колоссальными по своему объему и потребуют очень много ресурсов. Если у вас есть основания полагать, что набор данных состоит больше чем из нескольких строк, то следует предпринять действия по ограничению области действия запроса.



Дополнительная информация о том, как писать операторы SQL наподобие `SELECT`, `INSERT` и т.д. приведена в главе 13. Следует помнить, что одним из лучших способов исправления ошибок, возникающих при использовании операторов SQL в коде PHP, состоит в опробовании этих операторов (после присваивания подходящих фиктивных значений переменным) с использованием клиента MySQL с интерфейсом командной строки, а не клиента PHP. Более подробные сведения об отладке операторов SQL, применяемых в коде PHP, приведены в главе 19.

Наконец, чтобы завершить сеанс работы с клиентской программой MySQL, можно воспользоваться командой `quit;`. И в этой команде точка с запятой является необязательной. После ввода указанной команды вы снова возвратитесь к обычному приглашению командного интерпретатора.

Администрирование учетных записей MySQL

Важным фактором безопасной и эффективной эксплуатации MySQL является правильное применение системы прав доступа, предусмотренной в этой базе данных, и надлежащее использование инструментальных средств, предназначенных для управления правами доступа пользователей.

База данных MySQL позволяет чрезвычайно точно определять права доступа различных пользователей, которые подключаются к базе данных с помощью клиентских программ, находящихся в самых различных точках системы. Права доступа подразделяются на четыре нисходящих уровня: глобальные, базы данных, таблицы и столбцы. Поэтому теоретически предусмотрена возможность регламентировать доступ конкретного пользователя для записи данных только в указанные столбцы указанных таблиц указанных баз данных на указанном сервере MySQL. Столь же легко можно, не задумываясь, предоставить любому пользователю базы данных, подключающемуся откуда угодно, такие же права, как пользователю базы данных `root` (но такая организация защиты доступа категорически не рекомендуется).

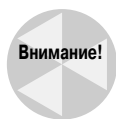
Очевидно, что согласно требованиям защиты обычно следует руководствоваться хорошим эмпирическим правилом — предоставлять каждому пользователю только минимальные права доступа, без которых он вообще не мог бы выполнять свои функции. Но при этом необходи-

мо учитывать и обратную сторону: чем более детализированной является применяемая схема прав доступа, тем медленнее выполняются все и каждые операторы INSERT, SELECT, UPDATE и DELETE. Причина этого состоит в том, что при наличии более подробно регламентированных прав доступа в базе данных MySQL приходится выполнять проверку большего количества таблиц с информацией о правах доступа. Разумеется, в действительности наблюдаемое при этом снижение производительности отрицательно отражается на работе далеко не каждого узла, но если такое влияние становится существенным, то необходимо найти определенный компромисс между степенью защиты и производительностью.

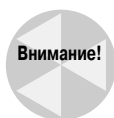
Основой системы прав доступа MySQL является таблица, с которой должен быть хорошо знаком каждый администратор базы данных — таблица user базы данных mysql (эта база данных, наряду с базой данных test, поставляется с каждой инсталляцией MySQL). Рассмотрим упрощенную версию указанной таблицы (приносим свои извинения за то, что строки этой таблицы показаны с разбивкой на две печатные строки, но так они отображаются и в окнах многих командных интерпретаторов).

```
mysql> select * from user;
+-----+-----+-----+-----+-----+
| Host      | User | Password | Select_priv | Insert_priv |
| Update_priv | Delete_priv |
+-----+-----+-----+-----+-----+
| localhost | root |          | Y           | Y           |
| Y         | Y   |          | Y           | Y           |
| dhcppc2   | root |          | Y           | Y           |
| Y         | Y   |          | Y           | Y           |
| localhost |      |          | N           | N           |
| N         | N   |          | N           | N           |
| dhcppc2   |      |          | N           | N           |
| N         | N   |          | N           | N           |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Как показывает данный листинг, в базе данных имеются данные о нескольких характерных глобальных правах доступа, которые обозначаются в таблице с помощью буквы Y или N. Буква Y в строке таблицы user обозначает глобальное право доступа, распространяющееся на все таблицы всех баз данных на рассматриваемом сервере MySQL. Если сервер MySQL получает запрос от пользователя, для которого задано значение N в поле, соответствующем действию, попытка выполнения которого предпринята пользователем, сервер приступает к прохождению вниз по иерархии уровней прав доступа. Вначале происходит переход к таблице db для проверки прав доступа на уровне базы данных. Затем, если в этой таблице также обнаруживаются только значения N, происходит переход к таблице tables_priv для ознакомления с правами доступа на уровне таблицы. Наконец, сервер переходит к таблице columns_priv для получения информации о правах доступа на уровне столбца. И только после исчерпывающей проверки всех таблиц с информацией о правах доступа сервер сообщает клиенту об ошибке доступа.



Даже если права доступа на уровне столбца или таблицы предоставлены лишь единственному пользователю среди многих, в системе MySQL осуществляется проверка указанных таблиц прав доступа для всех пользователей. Поэтому предоставление прав доступа к столбцу или таблице даже одному пользователю может существенно замедлить выполнение всех операторов SQL всех пользователей.



Не существует способа предоставить пользователю возможность создавать или уничтожать любую таблицу базы данных, не предоставив также этому пользователю возможность полностью уничтожить рассматриваемую базу данных. Тем не менее можно запретить пользователю создавать или уничтожать другие базы данных на том же сервере. Кроме того, таблицы с правами доступа MySQL нельзя использовать для блокировки попыток подключения с определенных IP-адресов или хостов.

Для добавления или редактирования прав доступа пользователей в базе данных MySQL могут применяться два разных способа (при условии, что модификацией прав доступа занимается пользователь базы данных `root`): непосредственное выполнение операторов SQL (например, ввод буквы `Y` вручную в каждое соответствующее поле каждой соответствующей таблицы прав доступа) или использование синтаксических конструкций `GRANT` и `REVOKE`. Последний способ является более легким и менее опасным, если допущена небольшая ошибка, поскольку в большинстве случаев попытка выполнения ошибочного запроса закончится неудачей с сообщением об ошибке SQL, но при этом не возникнет пробел в системе защиты.

Чтобы ввести информацию о новом пользователе MySQL, можно применить следующий оператор:

```
GRANT priv_type [(column1, column2, column3)]
ON database.[table]
TO user@host IDENTIFIED BY 'new_password';
```

Здесь данные о столбцах (`column`) и таблицах (`table`) являются необязательными, а с помощью списка, разделенного запятыми, могут быть заданы дополнительные сведения о типах прав доступа `priv_types`.

Типы прав доступа и области их действия показаны в табл. 14.1.

Таблица 14.1. Области действия прав доступа MySQL

Права доступа	Глобальные	Базы данных	Таблицы	Столбца
ALL	✓	✓		
ALTER	✓	✓	✓	
CREATE	✓	✓	✓	
CREATE TEMPORARY TABLE	✓	✓	✓	
DELETE	✓	✓	✓	
DROP	✓	✓	✓	
EXECUTE	✓	✓		
FILE	✓	✓		
INDEX	✓	✓		
INSERT	✓	✓	✓	✓
LOCK TABLES	✓	✓		
PROCESS	✓	✓		
REFERENCES	✓	✓		
RELOAD	✓	✓		
REPLICATION CLIENT	✓			
REPLICATION SLAVE	✓			
SELECT	✓	✓	✓	✓

Права доступа	Глобальные	Базы данных	Таблицы	Столбца
SHOW DATABASES	✓			
SHUTDOWN	✓			
SUPER	✓	✓		
UPDATE	✓	✓	✓	✓
USAGE	✓	✓		
GRANT OPTION	✓	✓	✓	

Очевидно, что нет смысла пытаться предоставить какому-либо пользователю права доступа SHUTDOWN на уровне таблицы. При этом просто выработывается сообщение об ошибке, в котором предлагается ознакомиться с документацией. Если предоставлены права доступа ALL на уровне столбца, таблицы или базы данных, то пользователь получает возможность применять только тот набор прав доступа, который соответствует указанному уровню.

Необходимо соблюдать исключительную осторожность при предоставлении пользователям следующих прав доступа, поскольку эти права все без исключения являются опасными: GRANT, ALTER, CREATE, DROP, FILE, SHUTDOWN, PROCESS. На производстве такие права доступа не требуются ни одному обычному пользователю базы данных, особенно пользователю PHP.

Синтаксическая конструкция оператора отмены прав доступа является весьма похожей на соответствующий оператор предоставления прав доступа, но проще него:

```
REVOKE priv_type [(column1, column2, column3)]
ON database[.table]
FROM user@host;
```

После предоставления или отмены прав доступа для любого пользователя необходимо вынудить базу данных выполнить перезагрузку в память новых данных о правах доступа. Для этого требуется ввести команду FLUSH PRIVILEGES. Можно также остановить и снова запустить сервер, но такое решение во многих обстоятельствах является практически неприменимым.

Безусловно, изложенные выше сведения вполне доступны для восприятия, но не дают ответа на такой вопрос: “Какие же права доступа должны быть фактически предоставлены действующим пользователям PHP?” Рассмотрим некоторые случаи, которые часто встречаются на практике.

Локальный сервер, применяемый для разработки

Если доступ осуществляется исключительно локально, особенно с компьютера, который не подключен постоянно к Интернет или надежно спрятан за хорошим брандмауэром, то применимы практически любые права доступа. Если в ходе разработки требуется проводить эксперименты со схемой базы данных, то наиболее подходящим является именно такая конфигурация, поэтому разработчику, кроме обычных прав доступа для выполнения операций SELECT, INSERT и UPDATE, могут быть предоставлены права доступа наподобие ALTER, CREATE, DELETE и DROP. При таких обстоятельствах многие администраторы считают приемлемым предоставить права доступа ALL PRIVILEGES к определенной базе данных локальному пользователю, как показано ниже.

```
GRANT ALL PRIVILEGES on database.*
TO username@localhost
IDENTIFIED BY 'password';
```

Автономный Web-узел

База данных, находящаяся на отдельном хосте, по-видимому, должна будет принимать запросы на установление соединений от многочисленных Web-серверов, находящихся в том же домене. На производстве для всех серверных компьютеров должны предоставляться лишь права доступа SELECT, INSERT, UPDATE и, возможно, DELETE, хотя во многих системах удаление данных фактически не происходит, поэтому уровень безопасности немного повышается, если право доступа DELETE не предоставляется. Кроме того, поскольку количество баз данных, подключающихся к производственной базе данных автономного Web-узла, по-видимому, не слишком велико, то применение глобальных прав доступа приводит к ускорению работы и вместе с тем не создает более значительный реальный риск нарушения защиты. Поэтому допустимый оператор предоставления прав доступа может выглядеть следующим образом:

```
GRANT SELECT, INSERT, UPDATE ON *.*  
TO phpdbuser@%.example.com  
IDENTIFIED BY 'password';
```

Тем не менее в подобной ситуации чаще всего используется также репликация по принципу “ведущий–ведомый”. Часто подобные кластеры баз данных MySQL имеют такую конфигурацию, что все запросы на выполнение операций записи поступают в ведущую базу данных, но ведомые базы данных не выполняют никаких других действий, кроме очень быстрого обслуживания операций чтения. В таком случае в каждой ведомой базе данных предоставляются только права доступа SELECT, а в ведущей базе данных предоставляются только права INSERT и UPDATE; при этом, возможно, эти права назначаются двум разным пользователям базы данных.

Web-узел на разделяемом хосте

Для провайдера Интернет, предоставляющего разделяемый хостинг, или для заказчика, разместившего свой Web-узел на разделяемом хосте, важнее добиться высокой степени защиты, чем высокой производительности. Ни при каких обстоятельствах нельзя допустить, чтобы один из пользователей имел возможность манипулировать данными или удалять данные, принадлежащие другому пользователю.

Если для каждого пользователя не предусмотрен свой собственный экземпляр MySQL, работающий через отдельный порт, то администратор провайдера Интернет, предоставляющего разделяемый хостинг, не должен разрешать пользователям выполнять глобальные операции создания или удаления объектов базы данных. Очевидно, что не существует удобного способа запретить создание или удаление таблиц, поскольку (как было описано выше) наличие у пользователя таких прав влечет за собой то, что пользователь получает также возможность удалить при желании свою собственную базу данных. Дело обстоит именно так — если пользователи имеют право определять новые таблицы, а такого права в указанной ситуации их, безусловно, нельзя лишать, то не остается приемлемого способа защитить пользователей от ошибки, в результате которой могут быть уничтожены все их данные с помощью единственной команды! В этом заключается обратная сторона удобства создания объектов в базе данных MySQL. Но администратор базы данных может и даже обязан предотвратить ситуацию, в которой пользователи получили бы возможность проделывать такие же операции с данными других пользователей на том же сервере.

Провайдеры Интернет не должны использовать подстановочные символы для обозначения имен пользователей или имен хостов. Для каждой учетной записи пользователя должно быть предусмотрено подключение с одного и только одного сервера в пространстве имен до-

мена. Для применения способа подключения, предусматривающего прием клиентских запросов MySQL на установление соединения из-за пределов брандмауэра, нет никаких приемлемых оснований. Дело в том, что пользователям для проведения манипуляций с их базой данных должно быть вполне достаточно предоставить возможность подключения с помощью командного интерпретатора `ssh` к их разделяемому пространству хостинга; еще один вариант состоит в предоставлении пользователям возможности использовать какое-либо инструментальное средство с графическим интерфейсом. Поэтому в данной ситуации чаще всего применяется такой оператор предоставления прав доступа:

```
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER
ON database.*
TO user@servername.example.com
IDENTIFIED BY 'password';
```

Кроме того, на разделяемых хостах часто используется такой недопустимый метод: пользователю разрешают или даже требуют от него повторно применять имя пользователя и пароль домена в качестве имени пользователя и пароля MySQL. Такой метод является недопустимым и может стать причиной возникновения проблем, не поддающихся описанию. Поэтому, если ваш провайдер Интернет допускает применение такого метода, авторы рекомендуют немедленно перейти на другой узел, предоставляющий Web-хостинг.

Программа PHPMyAdmin

Безусловно, авторы рекомендуют ознакомиться с клиентской программой MySQL, имеющей интерфейс командной строки, и использовать ее в максимально возможной степени, но действительность такова, что многим пользователям PHP не нравится интерфейс командной строки и они сталкиваются с затруднениями при попытке визуально представить свои структурированные данные, рассматривая эти данные в крошечном окне командного интерпретатора с такими ужасными результатами разбивки на несколько строк. Кроме того, некоторые Web-хосты не предоставляют полного доступа к командному интерпретатору и поэтому позволяют пользователям администрировать и просматривать свои базы данных только через Web-интерфейс. В связи с этим потребность в использовании графических интерфейсов на основе Web для MySQL была и остается просто поразительной, несмотря на то, что при этом возникают многочисленные проблемы защиты. Если вы собираетесь применять одну из таких клиентских программ (наиболее широко применяемой является PHPMyAdmin, но имеется также несколько аналогичных пакетов), то должны по меньшей мере знать, как следует эксплуатировать эти программы наиболее безопасным способом.

Основная проблема, связанная с использованием программы PHPMyAdmin, состоит в том, что эксплуатация этой программы не позволяет лучше понять проблемы защиты MySQL и к тому же становится причиной возникновения некоторых дополнительных нарушений защиты. Дело в том, что необходимо обеспечить надлежащую поддержку не только системы прав доступа MySQL, описанной ранее, но и создать соответствующий уровень защиты Web. К сожалению, программа PHPMyAdmin не позволяет легко добиться качественной защиты Web.

Допустим, что у пользователя имеется учетная запись для Web-хостинга с доступом к командному интерпретатору одного из компьютеров, находящихся в некотором центре обработки данных. Если сеть эксплуатируется должным образом, то пользователю приходится регистрироваться на данном компьютере, подключаясь через зашифрованное соединение, такое как `ssh`. После этого пользователь обязан отдельно ввести имя пользователя и пароль базы данных, а также, возможно, имя хоста и номер порта на другом компьютере, чтобы подключиться к серверу базы данных MySQL, а сам сервер должен принимать только такие запросы на установление соединения, которые исходят от того конкретного компьютера, где имеется

учетная запись пользователя. Такая система не является идеальной (если речь идет о защите, то идеальных систем не существует), но вполне приемлема.

В отличие от этого, при эксплуатации программы РНРМуAdmin не обязательно даже требуется задавать имя пользователя или пароль. Каждый, кто знает о существовании конкретной инсталляции РНРМуAdmin, может попытаться получить к ней доступ с помощью браузера из любой точки в сети. Если же используется конфигурация, предусмотренная по умолчанию, то потенциальный нарушитель приобретает все права в системе, даже не заметив существования подсистемы авторизации. Более того, если вы не возьмете на себя обязанности по выполнению такой настройки конфигурации Web-сервера, чтобы он принимал только запросы на установление соединения по протоколу HTTPS, то соединение между браузером и Web-сервером будет незашифрованным, поэтому все данные будут передаваться в прямом и обратном направлениях через обычное соединение HTTP.

Так как же воспользоваться удобными средствами программы РНРМуAdmin, сведя риск к минимуму? Прежде всего, следует оценить свои потребности. Многие люди обращаются к программе РНРМуAdmin только в очень редких случаях, например, для первоначальной настройки схемы базы данных. Если ваши потребности состоят именно в этом, то для вас вообще нет необходимости устанавливать программу РНРМуAdmin на производственном сервере. Установите базу данных локально, выполните настройку схемы базы данных в полном соответствии с вашими потребностями, а затем создайте копию определения локальной базы данных в виде кода SQL, который может быть применен на производственном сервере. Это — очень легкий способ, который является намного более безопасным и сводит к минимуму время простоя производственных серверов. Для получения дополнительной информации о создании копий структуры и данных базы данных с помощью программы `mysqldump` обратитесь к разделу “Создание резервных копий” ниже в этой главе.

Единственным недостатком рассматриваемого метода является то, что задача внесения время от времени небольших изменений в схему базы данных, которая уже содержит данные, становится весьма сложной. Однако в этом случае можно предусмотреть применение программы РНРМуAdmin, но запретить доступ к ней на тот период, когда она не используется. Для этого можно просто задать права доступа к каталогу, где находится программа РНРМуAdmin, и к файлам в этом каталоге таким образом, чтобы были запрещены запросы на подключение, поступающие от пользователя, под именем которого работает Web-сервер (т.е. от пользователя `nobody`). А в следующий раз, когда потребуются прибегнуть к использованию РНРМуAdmin, достаточно будет просто за одну секунду переопределить права доступа, для того чтобы пользователь, олицетворяющий программу Web-сервера, снова мог читать эти файлы. Тем не менее на то время, когда разрешен доступ к Web-клиенту программы РНРМуAdmin, желательно также применять аутентификацию с помощью протокола `http` или `cookie`-файлов.

Еще одна мера защиты может предусматривать использование шифрования SSL в каталоге РНРМуAdmin. Такой способ организации защиты может стать решением проблемы, связанной с передачей незашифрованных конфиденциальных данных между Web-сервером и браузером. Дополнительная информация по этой теме приведена на указанных ниже Web-узлах.

- www.modssl.org (демон `httpd` сервера Apache).
- www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/windowsserver2003/proddocs/deployguide/iisdg_meanfmd.asp (сервер IIS 6).

Наиболее широко применяемый метод повышения уровня защиты состоит в использовании схем аутентификации программы РНРМуAdmin с помощью протокола `http` или `cookie`-файлов. Для этого требуется создание специальной учетной записи пользователя РНРМуAdmin, с помощью которой может осуществляться чтение таблиц прав доступа

MySQL, наряду с обычной учетной записью администратора базы данных. Метод с использованием протокола http может применяться только при работе с сервером Apache. Метод с использованием cookie-файлов предусматривает шифрование паролей перед их записью в cookie-файлы. Этот метод может применяться на Web-серверах, отличных от Apache, и является единственным методом, который обеспечивает полный выход из системы.

Для использования авторизации с помощью протокола http или cookie-файлов создайте учетную запись пользователя PHPMyAdmin, как показано ниже (после разархивирования или распаковки пакета PHPMyAdmin в корневом каталоге документов Web-сервера).

```
GRANT USAGE ON mysql.* TO 'pmauser'@'localhost'
  IDENTIFIED BY 'pmapassword';
GRANT SELECT (
  Host, User, Select_priv, Insert_priv, Update_priv,
  Delete_priv, Create_priv, Drop_priv, Reload_priv,
  Shutdown_priv, Process_priv, File_priv, Grant_priv,
  References_priv, Index_priv, Alter_priv, Show_db_priv,
  Super_priv, Create_tmp_table_priv, Lock_tables_priv,
  Execute_priv, Repl_slave_priv, Repl_client_priv
) ON mysql.user TO 'pmauser'@'localhost';
GRANT SELECT ON mysql.db TO 'pmauser'@'localhost';
GRANT SELECT ON mysql.host TO 'pmauser'@'localhost';
GRANT SELECT (Host, Db, User, Table_name, Table_priv,
  Column_priv)
ON mysql.tables_priv TO 'pmauser'@'localhost';
```

При использовании обеих схем авторизации необходимо также задать значения следующих полей в файле `config.inc.php` программы PHPMyAdmin:

```
$cfg['PmaAbsoluteUri'] = 'http://localhost/phpMyAdmin';
$cfg['Servers'][$i]['host'] = 'localhost';
$cfg['Servers'][$i]['auth_type'] = 'http or cookie';
$cfg['Servers'][$i]['user'] = 'pmauser';
```

При использовании авторизации на основе cookie-файлов необходимо также задать значение следующего поля:

```
$cfg['blowfish_secret'] = 'Supersecret passphrase';
```

Теперь при попытке подключения появляется либо основное всплывающее окно авторизации Apache, либо Web-форма. Введите имя пользователя и пароль администратора базы данных (не используйте учетную запись пользователя базы данных root!), и вам будут предоставлены точно такие же права доступа MySQL, как и при использовании клиентской программы MySQL с интерфейсом командной строки.

Наконец, предусмотрен еще один сомнительный метод “аутентификации” PHPMyAdmin, который обычно не следует использовать, — так называемый метод с применением файла `config`. Этот метод заключается в том, что имя пользователя и пароль администратора базы данных вводятся в файл `config.inc.php`. После этого возможность просматривать базу данных получает любой, в чьем распоряжении имеется браузер. Единственными обстоятельствами, в которых можно было бы рассматривать возможность использования данного метода, является локальный компьютер, который не принимает запросы на установление соединений по протоколу HTTP из внешнего мира.

Следует отметить, что после установки программы PHPMyAdmin можно легко получить доступ к обширной информации о базах данных MySQL. Выбрав базу данных из списка доступных баз данных, вы увидите главный экран базы данных, наподобие приведенного на рис. 14.1.

Пользователи графического интерфейса, по-видимому, не столкнутся с какими-либо сложностями при перемещении по интерфейсу программы PHPMyAdmin, начиная с этого

экрана. Но, к сожалению, в программе PHPMyAdmin используется немного не такая терминология для обозначения операций, как в клиентской программе MySQL: “structure”, а не “show columns”; “browse”, а не “select”; “export”, а не “dump”, но после приобретения небольшого опыта работы с этой программой все намного упрощается.

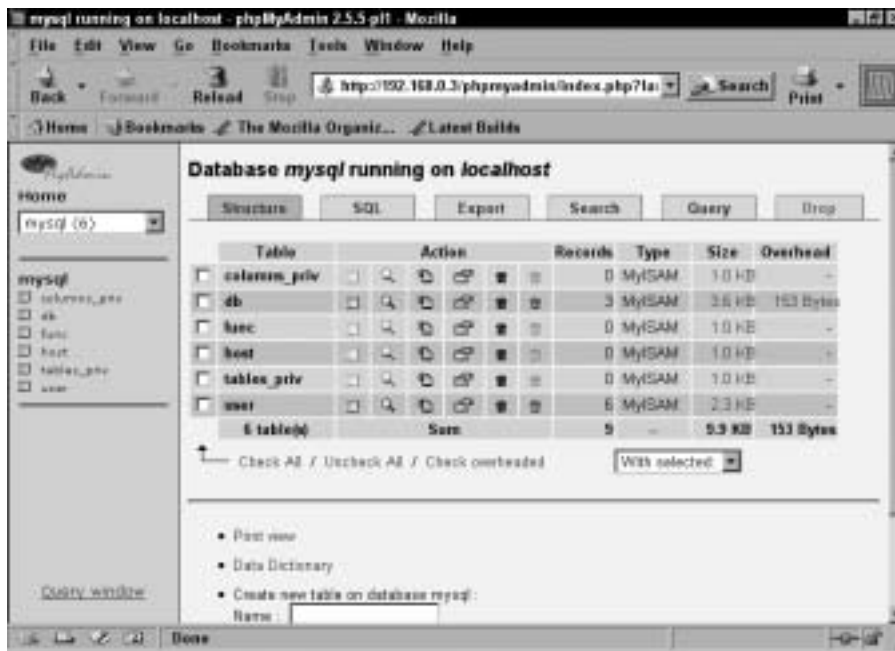
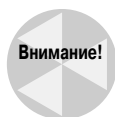


Рис. 14.1. Главный экран базы данных в программе PHPMyAdmin



Определенные сложности в работе с программой PHPMyAdmin возникают у людей с нарушением цветового зрения, поскольку все опасные операции обозначены с помощью пиктограмм или ссылок красного цвета. К счастью, прежде чем будет выполнена любая операция уничтожения или массового удаления, появляется запрос на подтверждение, поэтому вероятность возникновения неумышленных ошибок сводится к минимуму, при условии, что пользователь не нажимает клавишу <Enter> без разбора.

Создание резервных копий

Резервные копии баз данных можно создавать двумя способами: непосредственно копируя каталог с данными (либо вручную, либо с помощью сценария `mysqlhotcopy` в системе Unix) или используя инструментальное средство `mysqldump` для вывода информации в файл с командами SQL, который содержит копию базы данных. Первый способ немного быстрее, но последний предоставляет больше возможностей. С помощью утилиты `mysqldump` можно выбрать вариант копирования только структуры базы данных, только данных или того и другого.

Ниже приведен самый простой пример использования утилиты `mysqldump`.

```
mysqldump -u username -p databasename > dumpfilename.sql
```

Эта команда выводит текстовый файл, который может быть считан на другом сервере базы данных, примерно таким образом:

```
mysql -u root -p databasename < dumpfilename.sql
```

Вместо записи вывода утилиты `mysqldump` в файл полученные выходные данные можно также направить по каналу непосредственно на другой сервер, как показано ниже.

```
mysqldump -u username -p databasename |  
mysql -h remote-host -u remoteuser -p -C databasename
```

Тем не менее в некоторых случаях такой вариант вывода и ввода данных может оказаться менее безопасным, поскольку требует передачи удаленному хосту команды принимать запросы на установление соединений, в которых осуществляется модификация базы данных, от внешних клиентских программ.

Приведенная выше основная команда вполне оправдывает свое назначение. Под этим подразумевается, что выполнение указанной команды приводит к получению аккуратного файла с командами SQL, который содержит не только информацию о структуре, но и данные, полученные из рассматриваемой базы данных. Но иногда требуется дополнительно конкретизировать результаты, предоставляемые утилитой `mysqldump`, в связи с тем, что должна быть получена только информация о структуре или только данные. Другие опции позволяют получить содержимое всех баз данных на указанном сервере или только некоторых таблиц из выбранной базы данных. Сервер MySQL позволяет, с одной стороны, задавать различные комбинации баз данных и (или) таблиц, а с другой — указывать значения всевозможных опций в команде копирования.

Если для получения копии необходимо выбрать определенные таблицы из указанной базы данных, достаточно перечислить имена этих таблиц после имени базы данных:

```
mysqldump -u username -p databasename table1 table2  
> dumpfilename.sql
```

Если же требуется получить копию лишь некоторых баз данных, работающих под управлением конкретного сервера, то можно ввести опцию `--databases`, а затем задать список баз данных, как показано ниже. Но в этом случае отсутствует возможность указывать отдельные таблицы.

```
mysqldump -u username -p --databases database1 database2 >  
dumpfilename.sql
```

Если необходимо получить копию всех баз данных, то следует ввести опцию `--all-databases`:

```
mysqldump -u username -p --all-databases > dumpfilename.sql
```

Любую из описанных опций можно задать перед указанием имен баз данных и таблиц. Утилита `mysqldump` поддерживает широкий перечень опций, но в табл. 14.2 перечислены лишь наиболее часто используемые опции.

Таблица 14.2. Опции утилиты `mysqldump`

Опция	Описание
<code>--add-locks</code>	Ввести в файл с операторами SQL команду блокировки таблицы для обеспечения более быстрого выполнения операций вставки данных в целевую таблицу. См. также <code>--opt</code>
<code>--add-drop-table</code>	Перезаписать определение каждой таблицы. При использовании этой опции необходимо соблюдать осторожность, поскольку может произойти непреднамеренное удаление данных! Если эта опция не используется, а таблица с тем же именем уже существует, то после обращения к целевой базе данных вырабатывается сообщение об ошибке

Опция	Описание
-a, --all	Эта опция задает сразу все возможные опции. При использовании этой опции необходимо соблюдать осторожность!
-c, --complete-insert	Использовать более полные операторы вставки, с названиями столбцов, а не просто считывать значения
--help	Вывести справочное сообщение с описанием опций
-l, --lock-tables	Заблокировать таблицы в исходной базе (базах) данных перед созданием копии
-n, --no-create-db	Не создавать базы данных с указанными именами, если они еще не существуют. Эта опция применяется по умолчанию, если задана опция --databases или --all-databases
-t, --no-create-info	Не создавать таблицы с указанными именами, если они еще не существуют
-d, --no-data	Формировать только операторы с определениями структуры указанной базы (баз) данных или таблиц
--opt	Применение этой опции равносильно применению опций --quick --add-drop-table --add-locks --extended-insert --lock-tables и обеспечивает наиболее быстрое создание копии по сравнению со всеми прочими возможными вариантами. Если могут возникать конфликты имен, то следует обязательно удалить существующие таблицы
-q, --quick	Не применять буферизацию
-r, --result-file=filename	Вывести результирующую копию в указанный файл. В операционной системе DOS формируются символы с обозначением конца строки в стиле Unix
-w, --where='condition'	Осуществлять выборку результатов с применением конструкции WHERE, заданной в одинарных кавычках

Утилита `mysqldump` является очень удобной в использовании, поэтому на ее основе может (и обязательно должен) быть составлен график регулярного резервного копирования. Для выполнения операций, предусмотренных графиком резервного копирования, очень удобно использовать задания для утилиты `cron`. Если изменения в данные вносятся относительно редко, то достаточно будет выполнять резервное копирование один раз в неделю или два раза в месяц, а на узле с достаточно высоким трафиком может потребоваться запланировать операцию резервного копирования на выполнение один раз в сутки.

Пользователи программы `PHPMyAdmin` получают доступ к утилите `mysqldump` с помощью вкладки **Export**. Однако в настоящее время программа `PHPMyAdmin` позволяет использовать для получения копии данных только наиболее простые опции. А если требуется больший контроль над форматом файла с командами SQL, то следует вместо этого непосредственно использовать утилиту `mysqldump`, как описано выше.

Репликация

Метод репликации, применяемый в базе данных MySQL, основан на модели односторонней репликации по принципу “один ведущий—один или несколько ведомых”. Ведущая база данных обеспечивает выполнение всех операций записи. Под этим подразумевается вы-

полнение всех операций INSERT, UPDATE и DELETE, а также внесение всех изменений в схему. Ведомые базы данных периодически получают от ведущей базы данных и вносят в свои таблицы информацию об этих изменениях, а в интервалах между такими операциями модификации остаются доступными для использования в качестве высоко оптимизированных средств предоставления данных, предназначенных только для чтения (под этим подразумевается, что в эти интервалы ведущие базы данных выполняют только операции SELECT). В своей работе ведущая база данных никоим образом не учитывает существование ведомых баз данных, а просто вносит записи в двоичные журналы и предоставляет доступ к этим журналам. Все остальные действия выполняют ведомые базы данных: в них сопровождается график обновлений, осуществляется подключение к ведущей базе данных, считывается информация об изменениях, эти изменения вносятся в таблицы и т.д. Таким образом, ведомые базы данных имеют в своем распоряжении идентификационную информацию, которая описывает ведущую базу данных, а ведущая база данных не имеет сведений о наличии ведомых баз данных.

Если ведущая база данных по какой-либо причине прекращает свою работу, то выбор другой базы данных для ее замены не происходит автоматически. Чаще всего вся система становится неработоспособной, поскольку ведомые базы данных затрачивают много ресурсов, безуспешно пытаясь подключиться к ведущей базе данных для получения обновлений, в то время как система РНР предпринимает попытки выполнить операции чтения, но сделать этого не удается. При таких обстоятельствах администратор базы данных должен разорвать существующие связи “ведущий–ведомый” вручную и назначить новую ведущую базу данных. К счастью, в случае возникновения каких-либо нарушений в работе ведущей базы данных исправное состояние ведомых баз данных не может быть нарушено никоим образом, поэтому, при условии, что администратор базы данных своевременно обнаружит возникновение указанной проблемы и сможет с ней справиться, переход на новую ведущую базу данных должен произойти относительно быстро.

В каждой новой версии MySQL появляются многочисленные изменения и обновления, касающиеся репликации, поэтому одни современные версии часто несовместимы с другими современными версиями, выпущенными немного раньше или позже, в той части, которая относится к настройке средств репликации. В связи с этим основная рекомендация для тех, кто желает опробовать средства репликации, состоит в следующем: необходимо обеспечить, чтобы все используемые серверы базы данных MySQL имели одну и ту же версию. Более того, это должна быть версия 4.0.3 или последующая. Если же будет предпринята попытка организовать репликацию с использованием несовместимых версий MySQL (начиная с 3.23 и заканчивая 4.0.3), то создаваемая система репликации, по всей вероятности, не будет работать должным образом.

По существу, операции, которые должны быть выполнены для организации системы репликации на основе MySQL, состоят в следующем.

1. Предоставить необходимые права доступа к ведущей базе данных тому пользователю, от имени которого работают ведомые базы данных.
2. Получить снимок данных, находящихся в ведущей базе данных; скопировать эти данные на те компьютеры, где находятся ведомые базы данных.
3. Остановить серверы MySQL.
4. Перезапустить серверы MySQL после задания правильных значений параметра `server-id`.
5. Установить должным образом связь “ведущий–ведомый” от каждой ведомой базы данных.

Ниже этот процесс описан более подробно.

В ведущей базе данных необходимо создать учетную запись, предназначенную для использования ведомыми базами данными, и предоставить этой учетной записи права доступа REPLICATE SLAVE, как показано ниже. Какие-либо иные права доступа этой учетной записи предоставлять не требуется.

```
GRANT REPLICATE SLAVE ON *.*  
TO replicant@'%' IDENTIFIED BY 'replpwd';
```

Затем необходимо заблокировать ведущий сервер и получить снимок его состояния непосредственно перед репликацией. На ведущем сервере зарегистрируйтесь в клиентском сеансе MySQL в качестве пользователя root и выполните следующие команды:

```
FLUSH TABLES WITH READ LOCK;  
SHOW MASTER STATUS;
```

Выполнение этих команд приведет к тому, что внесение каких-либо изменений в базу данных станет невозможным до тех пор, пока вы не будете готовы перевести в рабочее состояние весь кластер, который состоит из ведущей и ведомых баз данных. Вы можете также (в зависимости от того, ведется ли эксплуатация данного сервера в режиме ведения журналов в двоичном формате) ознакомиться с некоторыми данными о местонахождении файла двоичного журнала и смещении. Если такие данные имеются, то запишите их на бумаге, а в противном случае используйте предусмотренные по умолчанию значения, соответственно ' ' (пустая строка) и 4.

Затем скопируйте структуру и данные ведущей базы данных. Для выполнения этой операции можно воспользоваться двумя способами. Первый из них состоит в копировании каталога mysql/data в файл формата tar или файл формата zip с помощью одной из следующих команд (или процедуры с графическим интерфейсом):

```
tar -cvf master_snapshot.tar data/  
zip master_snapshot.zip data/
```

Второй способ состоит в том, что может быть создана резервная копия с помощью утилиты mysqldump, как описано в следующем разделе. Скопируйте полученный файл снимка на каждый ведомый сервер.

Теперь остановите все серверы, ведущий и ведомый (ведомые). Закройте все сеансы клиентского командного интерпретатора mysql и введите следующую команду остановки на каждом сервере:

```
mysqladmin -u root -p shutdown
```

Необходимость в останове серверов связана с тем, что серверам, участвующим в репликации, должны быть присвоены уникальные значения параметра server-id. Серверы используют эти значения, чтобы найти друг друга в ходе установления связей “ведущий-ведомый”. Значение server-id задается в файле my.cnf каждого сервера и считывается при запуске. В операционной системе Windows файл my.cnf может находиться либо в C:\my.cnf, либо в C:\[Windows directory]\my.ini. В операционных системах типа Unix глобальный файл my.cnf находится по адресу /etc/my.cnf, а файл, относящийся к конкретному серверу (а при настройке системы репликации должен использоваться именно этот файл), находится по адресу /path/to/mysql/data/my.cnf.

Вначале необходимо задать значение server-id на компьютере ведущего сервера. Найдите или создайте файл my.cnf в том каталоге, который должен применяться для размещения этого файла на конкретной платформе, и убедитесь в том, что этот файл содержит такие строки:

```
[mysqld]  
log-bin  
server-id=1
```

Перезапустите ведущий сервер следующим образом:

```
bin/mysqld_safe --user=mysql
```

В файле `my.cnf` каждого ведомого сервера требуется только строка `server-id`, но не строка `log-bin`. Наиболее важное требование состоит в том, чтобы все значения `server-id` в каждом конкретном кластере репликации были уникальными! Это требование является категорическим, и если оно не соблюдается, то возникают серьезные нарушения в работе. Поэтому файл `my.cnf` первого ведомого сервера должен содержать такие строки:

```
[mysqld]
server-id=2
```

В таком случае в файле второго ведомого сервера можно задать `server-id=3` и т.д.

Теперь, перед переводом в рабочее состояние каждого ведомого сервера, необходимо выполнить небольшую вспомогательную работу. Если какой-то сервер MySQL уже использовался в качестве ведомого, необходимо удалить файлы `data/master.info` и `data/relay-log.info`. Кроме того, в каталоге `data` требуется удалить файлы `.err` и `.pid`. К тому же, если снимок данных ведущей базы данных скопирован на компьютер ведомой базы данных в виде файла формата `tar` или `zip`, то на данном этапе необходимо развернуть этот файл с помощью одной из следующих команд (эту команду необходимо выполнить после перехода в каталог `mysql`):

```
tar -xvf master_snapshot.tar
unzip master_snapshot.zip
```

Если же вместо этого применялась утилита `mysqldump`, то необходимо подождать, пока сервер не будет запущен.

Теперь выполните запуск ведомого сервера следующим образом:

```
bin/mysqld_safe --user=mysql --skip-slave-start --log-warnings
```

Если снимок данных ведущей базы данных был получен с помощью утилиты `mysqldump`, то на данном этапе можно применить полученный файл с командами SQL к ведомой базе данных, как показано ниже.

```
mysql -u root -p databasename < master_snapshot.sql
```

Наконец, необходимо установить связь “ведущий–ведомый”. Зарегистрируйтесь в командном интерпретаторе `mysql`, а затем введите следующие команды, подставляя значения, записанные на бумаге в начале процесса настройки:

```
CHANGE MASTER TO
  MASTER_HOST='masterhostname',
  MASTER_USER='replicant',
  MASTER_PASSWORD='replpwd',
  MASTER_LOG_FILE='',
  MASTER_LOG_POS=4;
START SLAVE;
```

Если возникают какие-либо проблемы, то с информацией о них можно ознакомиться, просмотрев журнал ошибок ведомого сервера.

Восстановление

В обычных условиях база данных MySQL не требует большого внимания. Серверы MySQL успешно работают в течение многих месяцев или даже лет при минимальном объеме администрирования. Однако с данными тоже возникают неприятности — ломаются жесткие

диски, на центрах хостинга внезапно пропадает питание, а ошибки людей всегда остаются постоянной и неизбежной угрозой. Если же на сервере не хватает оперативной памяти для всех эксплуатируемых приложений или становится недостаточным объем дискового пространства в разделе, отведенном для базы данных, также может возникнуть ошибка, которая потребует проведения процесса восстановления. Следует признать, что в ходе эксплуатации сервера MySQL, по сравнению с более мощными базами данных, вероятно, чаще возникают события, связанные с небольшими искажениями данных в базе данных. А возможно, просто администратору базы данных MySQL легче обнаружить такие события.

К счастью, база данных MySQL спроектирована таким образом, что позволяет удивительно легко исправить небольшие дефекты в данных и через короткое время снова перевести базу данных в рабочее состояние. Только однажды нам пришлось фактически снова создать всю базу данных из резервной копии после неудачного завершения многократных попыток восстановления, притом что сама авария была вызвана полным отказом жесткого диска. Но в предвидении такой ситуации разработчик не может составить продуктивный план или подготовить изящную процедуру восстановления, в которых бы не учитывалось требование по своевременному созданию резервных копий.

В состав программного обеспечения MySQL уже в течение длительного времени входит инструментальное средство с интерфейсом командной строки, называемой `myisamchk`, которое предназначено для проверки и исправления таблиц. Это — удобный сценарий, но отличается одним недостатком — его эффективное использование возможно, только если база данных остановлена. Этот недостаток остается вполне терпимым, если с помощью сценария `myisamchk` фактически осуществляется восстановление после аварии, поскольку так или иначе запуск базы данных с неисправными таблицами становится невозможным, но если требуется обеспечить предотвращение возможных проблем путем регулярной проверки таблиц с данными, то указанный недостаток становится весьма существенным. К счастью, в настоящее время имеется новое инструментальное средство проверки, которое может использоваться в ходе эксплуатации сервера, — `mysqlcheck`. С другой стороны, можно продолжать применять `myisamchk` в тот период, когда сервер не работает.

Оба эти указанные инструментальные средства по существу выполняют такие действия: проверяют таблицу MyISAM на наличие ошибок, исправляют обнаруженные ошибки и оптимизируют базу данных. Но для вызова соответствующих сценариев на выполнение применяются разные синтаксические конструкции.

Утилита `myisamchk`

Для вызова утилиты `myisamchk` применяется примерно такая команда:

```
myisamchk [options] table_name
```

или такая:

```
myisamchk [options] /path/to/mysql/data/database/table.MYI
```

Вместо имен каталогов базы данных `database` и имен таблиц `table` можно использовать подстановочный символ — звездочку, причем такой формат команды фактически применяется чаще, чем формат с указанием имени таблицы, поскольку обычно нельзя заранее определить, какая таблица является источником проблем. Для проверки всех таблиц во всех базах данных на серверном компьютере используются следующие команды:

```
myisamchk [options] /path/to/mysql/data/*/*.MYI
```

```
myisamchk [options] /path/to/mysql/data/*/*.MYD
```

Расширение `.MYI` относится к индексным файлам, а расширение `.MYD` — к файлам данных; необходимо проводить проверку тех и других.

Если опции не заданы, то утилита `myisamchk` просто проверяет указанную таблицу. Если же указана опция `-r`, то утилита `myisamchk` восстанавливает указанные таблицы. С помощью этой утилиты можно также проверить и восстановить любые поврежденные таблицы в одной операции:

```
myisamchk --silent --force --fast --update-state -O
key_buffer=64M -O sort_buffer=64M -O read_buffer=1 -O
write_buffer=1M /path/to/mysql/data/*/*.MYI
```

Кроме того, команда `myisamchk -r tablename` оптимизирует структуру таблицы, фрагментированной в результате выполнения операций удаления и обновления.

Утилита `mysqlcheck`

Новое инструментальное средство `mysqlcheck` обладает несколькими важными преимуществами перед `myisamchk`. Как было указано выше, утилита `mysqlcheck` может использоваться в ходе эксплуатации сервера, причем даже во время обслуживания запросов. Действие утилиты `mysqlcheck` распространяется на целые базы данных, а не на отдельные таблицы. Для вызова `mysqlcheck` на выполнение используются в основном такие же синтаксические конструкции, как и для вызова инструментального средства `mysqldump`, за одним исключением: исполняемый файл утилиты `mysqlcheck` можно скопировать и обозначить другим именем, чтобы тем самым заранее указать область применения утилиты. Утилита определяет, под каким именем она вызвана, и действует соответствующим образом.

Для вызова инструментального средства `mysqlcheck` на выполнение применяется один из следующих способов:

```
mysqlcheck [options] databasename table1 table2 table3
mysqlcheck [options] --databases database1 database2
mysqlcheck [options] --all-databases
```

Чтобы определить область применения утилиты как предназначенной для восстановления, анализа или оптимизации базы данных, достаточно скопировать файл `mysqlcheck` под именем `mysqlrepair`, `mysqlanalyze` или `mysqloptimize`, а затем вызывать эти файлы обычным образом, но уже не задавая опцию `-r`, `-a` или `-o`. Так, например, чтобы восстановить все базы данных, работающие под управлением конкретного сервера, достаточно ввести следующую команду:

```
mysqlrepair -u root -p --all-databases
```

Компания MySQL AB рекомендует установить регулярный график проверки файлов данных с применением заданий `cron`, а на этапе запуска сервера MySQL каждый раз вызывать на выполнение одну из указанных утилит. Соблюдение этой рекомендации позволит поддерживать компактную структуру хранения данных, способствующую ускорению операций чтения; заблаговременно устранять незначительные ошибки в данных, которые могут со временем привести к серьезным проблемам; уменьшить вероятность возникновения таких нарушений в ходе эксплуатации базы данных, которые могли бы повлиять на работу пользователей.

Резюме

MySQL относится к числу баз данных, обеспечивающих наиболее простое администрирование. Освоение навыков работы с этой базой данных может принести разработчикам PHP большую пользу. В последнее время процедура инсталляции MySQL на многих платформах значительно упростилась. Кроме того, в распоряжении пользователя имеется широкий перечень

инструментальных средств администрирования с графическим интерфейсом и с интерфейсом командной строки, предназначенных для просмотра структуры базы данных, управления пользователями базы данных и создания резервных копий. А что касается более сложных задач администрирования, к которым относится восстановление после сбоя и репликация, то база данных MySQL позволяет решать обе эти задачи предельно просто. Однако в последнее время произошли серьезные изменения, касающиеся условий совместного использования программных продуктов MySQL и PHP. В частности, изменились принципы лицензирования, стали несовместимыми версии клиентских программ, выпущено новое расширение `mysqli` и появилась поддержка транзакций. С информацией об этом необходимо ознакомиться не только начинающим пользователям MySQL, но и пользователям с большим стажем.