МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

НАБЕРЕЖНОЧЕЛНИНСКИЙ ИНСТИТУТ

СУБД MYSQL

Учебно-методическое пособие
по дисциплине
«ВЕБ-ПРОГРАММИРОВАНИЕ»

Набережные Челны
2018

Галиуллин Л.А. СУБД MySQL: учебно-методическое пособие по дисциплине «Веб-программирование» [Электронный ресурс] / Казанский федеральный университет, Электронный архив, 2018.

Рассматривается проблемы доступа к базам данных и СУБД MySQL. Представлены система безопасности MySQL, таблицы базы данных MySQL, рассмотрена работа с СУБД. Приведены контрольные вопросы. Для студентов направлений подготовки «Информатика и вычислительная техника», «Программная инженерия».

Введение

Самыми простыми базами данных изо всех возможных являются плоские текстовые файлы. Для доступа к ним можно использовать средства DHTML либо файловые операции, включенные в состав серверных языков сценариев. Для взаимодействия с популярными настольными базами данных, такими как хорошо известная СУБД Microsoft Access, используются свои технологии. Поскольку самыми популярными настольными операционными системами по-прежнему остаются представители семейства Microsoft Windows, средства доступа к настольным СУБД будут рассмотрены в пособии, посвященном серверному языку сценариев Active Server Pages (ASP). Самыми же продвинутыми средствами доступа располагают серверы баз данных, такие как MS SQL или MySQL, особенностям которого и посвящена данное пособие.

СУБД MySQL

MySQL является, возможно, самым ярким программным проектом после выхода Linux. Сейчас она серьезный конкурент большим СУБД в области разработки баз данных малого и среднего масштаба. Особыми целями проектирования MySQL были скорость, надежность и простота использования. Чтобы достичь такой производительности, ее разработчик - шведская фирма TcX приняла решение сделать многопоточным внутренний механизм MySQL. Многопоточное приложение одновременно выполняет несколько задач - так, как если бы одновременно выполнялось несколько экземпляров приложения.

Сделав MySQL многопоточной, TcX дала пользователям много выгод. Каждое входящее соединение обрабатывается отдельным потоком, при этом еще один всегда выполняющийся поток управляет соединениями, поэтому клиентам не приходится ждать завершения выполнения запросов других клиентов. Одновременно может выполняться любое количество запросов. Пока какой-либо поток записывает данные в таблицу,

все другие запросы, требующие доступа к этой таблице, просто ждут, пока она освободится. Клиент может выполнять все допустимые операции, не обращая внимания на другие одновременные соединения. Управляющий поток предотвращает одновременную запись какими-либо двумя потоками в одну и ту же таблицу. Такая архитектура более сложна, чем однопоточная. Однако выигрыш в скорости благодаря одновременному выполнению нескольких запросов значительно превосходит потери скорости, вызванные увеличением сложности.

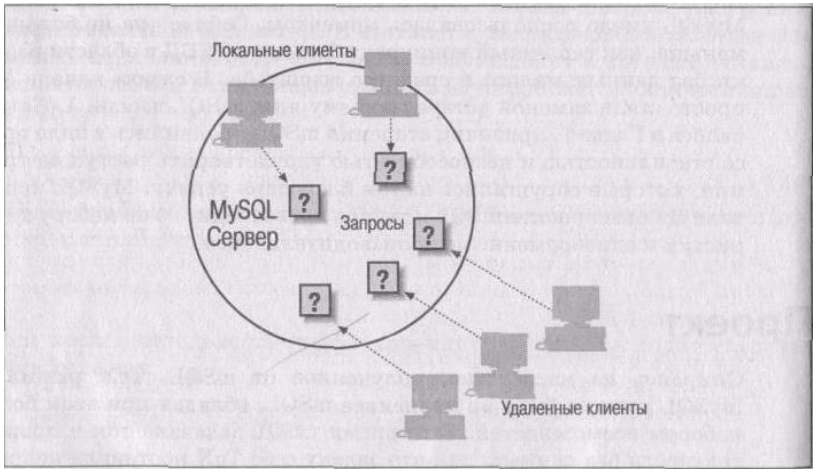

Рис. 1. Клиент-серверная архитектура MySQL

Другое преимущество многопоточной обработки присуще всем многопоточным приложениям. Несмотря на то, что потоки совместно используют память процесса, они выполняются раздельно. Благодаря этому разделению выполнение потоков на многопроцессорных машинах может быть распределено по нескольким ЦП. На рис. 1 показана эта многопоточная природа сервера MySQL.

Помимо выигрыша в производительности, полученного благодаря многопоточности, MySQL поддерживает большое подмножество языка запросов SQL. MySQL поддерживает более десятка типов данных, а также функции SQL. Ваше приложение

может получить доступ к этим функциям через команды ANSI SQL. MySQL фактически расширяет ANSI SQL несколькими новыми возможностями. В их числе новые функции (ENCRYPT, WEEKDAY, IF и другие), возможность инкрементирования полей (AUTO_INCREMENT и LAST_INSERT ID), а также возможность различать верхний и нижний регистры.

TcX намеренно опустила некоторые возможности SQL, встречающиеся в больших базах данных. Наиболее заметно отсутствие транзакций и встроенных процедур. TcX решила, что реализация этих возможностей нанесет слишком сильный удар по производительности. Однако TcX продолжает работу в этом направлении, но так, чтобы от потери производительности страдали только те пользователи, которым такие возможности действительно необходимы.

С 1996 года TcX использует MySQL в среде, где имеется более 40 баз данных, содержащих 10 000 таблиц. Из этих 10 000 более 500 таблиц имеют, в свою очередь, более 7 миллионов записей - около 100 Гбайт данных.

Система безопасности MySQL

Вам не только нужно иметь надежный доступ к своим данным, но и быть уверенным, что у других нет никакого доступа к ним. MySQL использует собственный сервер баз данных для обеспечения безопасности. При первоначальной установке MySQL создается база данных под названием «mysql». В этой базе есть пять таблиц: db, host, user, tables_priv, и columns_priv. Более новые версии MySQL создают также базу данных с названием func, но она не имеет отношения к безопасности. MySQL использует эти таблицы для определения того, кому что позволено делать. Таблица user содержит данные по безопасности, относящиеся к серверу в целом. Таблица host содержит права доступа к серверу для удаленных компьютеров. И наконец, db, tables_priv и columns_priv управляют доступом к отдельным базам данных, таблицам и колонкам.

Мы кратко рассмотрим все таблицы, поддерживающие безопасность в MySQL, а затем рассмотрим технологию их использования при обеспечении защиты ядром MySQL.

Таблица user

Таблица user имеет вид, показанный в Таблице 1:

Таблица 1. Таблица user

Поле	Тип	Null	Ключ	Значение по умолчанию
Host	char(60)		PRI	
User	char(16)		PRI	
Password	char(16)			
Select_priv	enum('N','Y')			N
Insert_priv	enum('N','Y')			N
Update_priv	enum('N','Y')			N
Delete_priv	enum('N','Y')			N
Create_priv	enum('N','Y')			N
Drop_priv	enum('N','Y')			N
Reload_priv	enum('N','Y')			N
Shutdown_priv	enum('N','Y')			N
Process_priv	enum('N','Y')			N
File_priv	enum('N','Y')			N
Grant_priv	enum('N','Y')			N
References_priv	enum('N','Y')			N
Index_priv	enum('N','Y')			N
Alter_priv	enum('N','Y')			N

В колонках Host и User можно использовать символ «%», заменяющий произвольную последовательность символов. Например, имя узла «chem%lab» включает в себя «chembiolab», «chemtestlab» и т. д. Специальное имя пользователя «nobody» действует как одиночный «%», то есть охватывает всех пользователей, не упомянутых где-либо в другом месте. Ниже разъясняется смысл различных прав доступа:

Select_priv - Возможность выполнять команды SELECT.

Insert_priv - Возможность выполнять команды INSERT.

Update_priv - Возможность выполнять команды UPDATE.

Delete_priv - Возможность выполнять команды DELETE.

Create_priv - Возможность выполнять команды CREATE или создавать базы данных.

Drop_priv - Возможность выполнять команды DROP для удаления баз данных.

Reload_priv - Возможность обновлять информацию о доступе с помощью *mysqladmin reload*.

Shutdown_priv - Возможность останавливать сервер через *mysqladmin shutdown*.

Process_priv - Возможность управлять процессами сервера.

File_priv - Возможность читать и записывать файлы с помощью команд типа SELECT INTO OUTFILE и LOAD DATA INFILE.

Grant_priv - Возможность давать привилегии другим пользователям.

Index_priv - Возможность создавать и уничтожать индексы.

Alter_priv - Возможность выполнять команду ALTER TABLE.

В MySQL есть специальная функция, позволяющая скрыть пароли от любопытных глаз. Функция password() зашифровывает пароль. Ниже показано, как использовать функцию password() в процессе добавления пользователей в систему.

INSERT INTO user (Host, User, Password, Select_priv, Insert_priv, Update_priv, Delete_priv)

VALUES ('%', 'bob', password('mypass'), 'Y', 'Y', 'Y', 'Y')

INSERT INTO user (Host, User, Password, Select_priv)

VALUES ('athens.imaginary.com', 'jane', '', 'Y')

INSERT INTO user(Host, User, Password)

VALUES ('%', 'nobody', '')

INSERT INTO user (Host, User, Password, Select_priv, Insert_priv, Update_priv, Delete_priv)

VALUES ('athens.imaginary.com', 'nobody', password('thispass'), 'Y', 'Y', 'Y', 'Y')

И

мена пользователей MySQL обычно не связаны с именами пользователей операционной системы. По умолчанию клиентские средства MySQL используют при регистрации имена пользователей операционной системы, однако, обязательного соответствия не требуется. В большинстве клиентских приложений MySQL можно с помощью параметра -u подключиться к MySQL, используя любое имя. Ваше имя

пользователя операционной системы не появится в таблице user, если не будет специально включено в нее с присвоением прав.

Первый созданный нами пользователь, «bob», может подключаться к базе данных с любого компьютера и выполнять команды SELECT, INSERT, UPDATE и DELETE. Второй пользователь, «jane», может подключаться с «athens.imaginary.com», не имеет пароля и может выполнять только SELECT. Третий пользователь - «nobody» - с любой машины. Этот пользователь вообще ничего не может делать. Последний пользователь - «nobody» - с машины «athens.imaginary.com», он может выполнять SELECT, INSERT, UPDATE и DELETE, как и пользователь «bob».

Как MySQL производит сопоставление? Некоторое имя может соответствовать на деле нескольким записям. Например, «nobody @athens.imaginary.com» соответствует и «nobody@%», и «nobody@athens.imaginary.com». Прежде чем осуществлять поиск в таблице user, MySQL сортирует данные следующим образом:

1. Сначала ищется соответствие для узлов, не содержащих масок «%», при этом пустое поле Host трактуется как «%».
2. Для одного и того же узла сначала проверяется соответствие имен, не содержащих масок. Пустое поле User трактуется как содержащее «%».
3. Первое найденное соответствие считается окончательным.

В предыдущем примере пользователь сначала будет сравниваться с «nobody» из «athens.imaginary.com», поскольку «athens.imaginary.com» в порядке сортировки стоит выше «%». Поскольку имена компьютеров сортируются раньше имен пользователей, значения привилегий для компьютера, с которого вы подключаетесь, имеют приоритет перед любыми конкретными правами, которые у вас могут быть. Если таблица user содержит записи:

Host	**User**
%	jane
athens.imaginary.com	

И jane подключается с «athens.imaginary.com», то MySQL

будет использовать привилегии, данные «athens.imaginary.com».

Таблица db

В таблице user не упоминаются конкретные базы данных и таблицы. Таблица user управляет сервером в целом. Однако на сервере обычно находится несколько баз данных, которые служат различным целям и, соответственно, обслуживают разные группы пользователей. Права доступа к отдельным базам данных хранятся в таблице db:

Таблица 2. Таблица db

Поле	Тип	Null	Ключ	Значение по умолчанию
Host	char(60)		PRI	
Db	char(32)		PRI	
User	char(16)		PRI	
Select_priv	enum('N','Y')			N
Insert_priv	enum('N','Y')			N
Update_priv	enum('N','Y')			N
Delete_priv	enum('N','Y')			N
Create_priv	enum('N','Y')			N
Drop_priv	enum('N','Y')			N
References_priv	enum('N','Y')			N
Index_priv	enum('N','Y')			N
Alter_priv	enum('N','Y')			N

Эта таблица во многом похожа на таблицу user. Основное отличие в том, что вместо колонки Password имеется колонка Db. Таблица управляет правами пользователей в отношении определенных баз данных. Поскольку привилегии, указанные в таблице user, относятся ко всему серверу в целом, права, присвоенные пользователю в таблице user, перекрывают права, присвоенные тому же пользователю в таблице. Например, если пользователю в таблице user разрешают доступ типа INSERT, это право действует в отношении всех баз данных, вне зависимости от того, что указано в таблице db.

Наиболее эффективно создание в таблице user записей для

всех пользователей, в которых не даны никакие права. В этом случае пользователь может лишь подключиться к серверу, не выполняя никаких действий. Исключение делается только для пользователя, назначенного администратором сервера. Все остальные должны получить права доступа через таблицу db. Каждый пользователь должен присутствовать в таблице user, иначе он не сможет подключаться к базам данных.

Те же правила, которые действуют в отношении колонок User и Host в таблице user, действуют и в таблице db, но с некоторой особенностью. Пустое поле Host вынуждает MySQL найти запись, соответствующую имени узла пользователя, в таблице host. Если такой записи не найдено, MySQL отказывает в доступе. Если соответствие найдено, MySQL определяет права как пересечение прав, определяемых таблицами host и db. Иными словами, в обеих записях разрешение должно иметь значение «Y», иначе в доступе отказывается.

Таблица host

Таблица host служит особой цели. Ее структура показана в таблице 3.3:

Таблица 3. Таблица Host

Поле	**Тип**	**Null**	**Ключ**	**Значение по умолчанию**
Host	char(60)		PRI	
Db	char(32)		PRI	
Select_priv	enum('N','Y')			N
Insert_priv	enum('N','Y')			N
Update_priv	enum('N','Y')			N
Delete_priv	enum('N','Y')			N
Create_priv	enum('N','Y')			N
Drop_priv	enum('N','Y')			N
Grant_priv	enum('N','Y')			N
References_priv	enum('N','Y')			N
Index_priv	enum('N','Y')			N
Alter_priv	enum('N','Y')			N

Таблица host позволяет задать основные разрешения на межкомпьютерном уровне. При проверке прав доступа MySQL ищет в таблице db соответствие имени пользователя и его машине. Если он находит запись, соответствующую имени пользователя, поле host которой пусто, MySQL обращается к таблице host и использует пересечение обоих прав для определения окончательного права доступа. Если у вас есть группа серверов, которые вы считаете менее защищенными, то вы можете запретить для них все права записи. Если «bob» заходит с одной из таких машин, и его запись в таблице db содержит пустое поле host, ему будет запрещена операция записи, даже если она разрешена ему согласно таблице db.

Таблицы tables_priv и columns_priv

Эти две таблицы, по сути, уточняют данные, имеющиеся в таблице db. Именно, право на всякую операцию сначала проверяется по таблице db, затем по таблице tables_priv , затем по таблице columns_priv. Операция разрешается, если одна из них дает разрешение. С помощью этих таблиц можно сузить область действия разрешений до уровня таблиц и колонок. Управлять этими таблицами можно через команды SQL GRANT и REVOKE.

Последовательность контроля доступа

Соединим элементы системы защиты MySQL вместе и покажем, как можно ими пользоваться в реальных ситуациях. MySQL осуществляет контроль доступа в два этапа. Первый этап - подключение. Необходимо подключиться к серверу, прежде чем пытаться что-либо сделать.

При подключении проводятся две проверки. Сначала MySQL проверяет, есть ли в таблице user запись, соответствующая имени пользователя и машины, с которой он подключается. Поиск соответствия основывается на правилах, которые мы обсудили раньше. Если соответствие не найдено, в доступе отказывается. В случае, когда соответствующая запись найдена

и имеет непустое поле Password , необходимо ввести правильный пароль. Неправильный пароль приводит к отклонению запроса на подключение.

Если соединение установлено, MySQL переходит к этапу верификации запроса. При этом сделанные вами запросы сопоставляются с вашими правами. Эти права MySQL проверяет по таблицам user, db, host, tables_pnv и columns_priv. Как только найдено соответствие в таблице user с положительным разрешением, команда немедленно выполняется. В противном случае MySQL продолжает поиск в следующих таблицах в указанном порядке:

1. db
2. tables_priv
3. columns_priv

Если таблица db содержит разрешение, дальнейшая проверка прекращается и выполняется команда. Если нет, то MySQL ищет соответствие в таблице tables_priv . Если это команда SELECT, объединяющая две таблицы, то пользователь должен иметь разрешения для обеих этих таблиц. Если хотя бы одна из записей отказывает в доступе или отсутствует, MySQL точно таким же способом проверяет все колонки в таблице columns_priv.

Изменение прав доступа

MySQL загружает таблицы доступа при запуске сервера. Преимуществом такого подхода по сравнению с динамическим обращением к таблицам является скорость. Отрицательная сторона состоит в том, что изменения, производимые в таблицах доступа MySQL, не сразу начинают действовать. Для того чтобы сервер увидел эти изменения, необходимо выполнить команду mysqladmin reload. Если таблицы изменяются с помощью SQL-команд GRANT или REVOKE, явно перегружать таблицы не требуется.

Утилиты MySQL

TcX распространяет MySQL с большим набором

вспомогательных утилит, однако набор утилит, предлагаемых сторонними разработчиками, еще богаче.

Утилиты командной строки (Command Line Tools)

Isamchk - Производит проверку файлов, содержащих данные базы. Эти файлы называются ISAM-файлами (ISAM — метод индексированного последовательного доступа). Эта утилита может устранить большую часть повреждений ISAM-файлов.

Isamlog - Читает создаваемые MySQL журналы, относящиеся к ISAM-файлам. Эти журналы можно использовать для воссоздания таблиц или воспроизведения изменений, внесенных в таблицы в течение некоторого промежутка времени.

mysql - Создает прямое подключение к серверу баз данных и позволяет вводить запросы непосредственно из приглашения MySQL.

mysqlaccess - Модифицирует таблицы прав доступа MySQL и отображает их в удобном для чтения виде. Использование этой утилиты - хороший способ изучения структуры таблиц доступа MySQL.

Mysqladmin - Осуществляет административные функции. С помощью этой утилиты можно добавлять и удалять целые базы данных, а также завершать работу сервера.

Mysqlbug - Составляет для TcX отчет о возникшей в MySQL неполадке. Отчет будет также послан в почтовый список рассылки MySQL, и армия добровольцев MySQL будет исследовать проблему.

Mysqldump - Записывает все содержимое таблицы, включая ее структуру, в файл в виде SQL-команд, которыми можно воссоздать таблицу. Выходные данные этой утилиты можно использовать для воссоздания таблицы в другой базе или на другом сервере. Синтаксис ее применения: *mysqldump -u user -p dbname --tab=path*, где path - путь для сохранения файлов.

Mysqlimport - Считывает данные из файла и вводит их в таблицу базы данных. Это должен быть файл с разделителями, где разделитель может быть любого обычного вида, например, запятая или кавычки.

Mysqlshow - Выводит на экран структуру баз данных,

имеющихся на сервере, и таблицы, из которых они состоят.

Контрольные вопросы

1. Что Вы знаете о СУБД?
2. Что Вы знаете о СУБД MySQL?
3. Что Вы знаете о системе безопасности MySQL?
4. Что Вы знаете о последовательности контроля доступа?
5. Что Вы знаете об изменении прав доступа?
6. Что Вы знаете об утилитах MySQL?
7. Что Вы знаете об утилитах сторонних разработчиков?
8. Что Вы знаете об интерфейсах CGI?
9. Что Вы знаете о клиентских приложениях?
10. Что Вы знаете об языке SQL?

Рекомендуемые источники

1. Колдаев В.Д. Основы алгоритмизации и программирования: Учебное пособие / В.Д. Колдаев; Под ред. Л.Г. Гагариной. - М.: ИД ФОРУМ: ИНФРА-М, 2015. - 416 с. [Электронный ресурс]. http://znanium.com/bookread.php?book=336649.
2. Гагарина Л.Г. Технология разработки программного обеспечения: Учеб. пос. / Л.Г.Гагарина, Е.В.Кокорева, Б.Д.Виснадул; Под ред. проф. Л.Г.Гагариной - М.: ИД ФОРУМ: НИЦ Инфра-М, 2017. - 400 с. [Электронный ресурс]. http://znanium.com/bookread.php?book=389963.
3. Голицына О. Л. Программирование на языках высокого уровня: Учебное пособие / О.Л. Голицына, И.И. Попов. - М.: Форум, 2016. - 496 с. [Электронный ресурс]. http://znanium.com/bookread.php?book=139428.