



---

# СУБД mySQL




# mysql

---

- MySQL - это торговая марка MySQL AB.
- Программное обеспечение MySQL (TM) представляет собой очень быстрый многопоточный, многопользовательский надежный SQL-сервер баз данных.
- Сервер MySQL предназначен как для критических по задачам производственных систем с большой нагрузкой, так и для встраивания в программное обеспечение массового распространения.
- Программное обеспечение MySQL имеет двойное лицензирование. Это означает, что пользователи могут выбирать, использовать ли ПО MySQL бесплатно по общедоступной лицензии GNU General Public License (GPL) или приобрести одну из стандартных коммерческих лицензий MySQL AB.

# Клиент-серверная технология



---

- ПО MySQL является системой клиент-сервер, которая содержит многопоточный SQL-сервер, обеспечивающий поддержку различных вычислительных машин баз данных, а также несколько различных клиентских программ и библиотек, средства администрирования и широкий спектр программных интерфейсов (API).
- Сервер MySQL также поставляется в виде многопоточной библиотеки, которую можно подключить к пользовательскому приложению и получить компактный, более быстрый и легкий в управлении продукт.

# Внутренние характеристики и переносимость

- Написан на C и C++. Протестирован на множестве различных компиляторов.
- Работает на различных платформах.
- Для обеспечения переносимости используется GNU Automake, Autoconf и Libtool.
- API для C, C++, Eiffel, Java, Perl, PHP, Python, Ruby и Tcl.
- Полностью многопоточный с использованием потоков ядра. Это означает, что, если такая возможность обеспечивается, можно легко организовать работу с несколькими процессорами.
- Очень быстрые дисковые таблицы на основе B-деревьев со сжатием индексов.
- Очень быстрая базирующаяся на потоках система распределения памяти.
- Очень быстрые соединения, использующие оптимизированный метод однопроходного мультисоединения (one-sweep multi-join).
- Хеш-таблицы в памяти, используемые как временные таблицы.
- SQL-функции реализованы при помощи хорошо оптимизированной библиотеки классов, поэтому они выполняются настолько быстро, насколько это возможно. Обычно после инициализации запроса распределения памяти не происходит вообще.
- MySQL- код протестирован с использованием Purify (коммерческий детектор утечки памяти), а также Valgrind, одного из GPL-инструментов.

# Поддерживаемые типы полей



---

- Большое количество: целочисленные со знаком/беззнаковые, длиной в 1, 2, 3, 4 и 8 байтов, FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET и ENUM.
- С записями фиксированной и переменной длины.
- Все столбцы имеют значения по умолчанию. С помощью INSERT можно вставить подмножество столбцов таблицы; столбцы, для которых явно не заданы значения, устанавливаются в значения по умолчанию.



# Команды и функции

---

- Полная поддержка операторов и функций в SELECT- и WHERE- частях запросов.
  - Например:
    - `mysql> SELECT CONCAT(first_name, " ", last_name)`
    - `-> FROM tbl_name`
    - `-> WHERE income/dependents > 10000 AND age > 30;`
- Полная поддержка для операторов SQL GROUP BY и ORDER BY с выражениями SQL. Поддержка групповых функций (COUNT(), COUNT(DISTINCT ...), AVG(), STD(), SUM(), MAX() и MIN()).
- Поддержка LEFT OUTER JOIN и RIGHT OUTER JOIN с синтаксисом ANSI SQL и ODBC.
- Разрешены псевдонимы для таблиц и столбцов в соответствии со стандартом SQL92.
- DELETE, INSERT, REPLACE, and UPDATE возвращают число строк, которые были изменены. Вместо этого можно задать возвращение совпавших строк. Для этого следует установить флаг при соединении с сервером.
- Команду SHOW, которая является специфической для MySQL, можно использовать для получения информации о базах данных, таблицах и индексах. Чтобы выяснить, как оптимизатор выполняет запрос, можно применять команду EXPLAIN.
- Имена функций не конфликтуют с именами таблиц и столбцов. Например, ABS является корректным именем столбца. Для вызова функции существует только одно ограничение: между именем функции и следующей за ним открывающей скобкой '(' не должно быть пробелов. В одном и том же запросе могут указываться таблицы из различных баз данных (с версии 3.22).

# Масштабируемость и ограничения



---

- Управляет очень большими базами данных. Компания MySQL AB использует MySQL для работы с несколькими базами данных, которые содержат 50 миллионов записей, кроме того, известны пользователи, использующие MySQL для работы с 60000 таблицами, включающими около 5000000000 строк.
- Для каждой таблицы разрешается иметь до 32 индексов. Каждый индекс может содержать от 1 до 16 столбцов или частей столбцов. Максимальная ширина индекса 500 бит (это значение может быть изменено при компиляции MySQL). Для индекса может использоваться префикс поля CHAR или VARCHAR.



# Размеры таблиц

---

- MySQL версии 3.22 имеет предел по размеру таблиц 4 Гб. В MySQL версии 3.23, где используется новый тип таблиц, максимальный размер таблицы доведен до 8 миллионов терабайтов ( $2^{63}$  bytes).
- Однако следует заметить, что операционные системы имеют свои собственные ограничения по размерам файлов.
- По умолчанию MySQL-таблицы имеют максимальный размер около 4 Гб.
- Для любой таблицы можно проверить/определить ее максимальный размер с помощью команд `SHOW TABLE STATUS` или `myisamchk -dv table_name`.





# Установка соединения

---

- Клиенты могут соединяться с MySQL, используя сокеты TCP/IP, сокеты Unix или именованные каналы (named pipes, под NT).
- Поддержка ODBC (Open-DataBase-Connectivity) для Win32 (с исходным кодом). Все функции ODBC 2.5 и многие другие. Например, для соединения с MySQL можно использовать MS Access.



# Лицензии MySQL

---

- ПО баз данных MySQL выпускается по общедоступной лицензии GNU General Public License (GPL).
- Это означает, что при соблюдении условий GPL ПО MySQL можно пользоваться бесплатно. Если вы не хотите связывать себя лицензионными ограничениями GPL (например, вас не устраивает условие, что ваше собственное приложение также подпадает под действие GPL), есть возможность приобрести у компании MySQL AB на этот же продукт коммерческую лицензию.

# Использование ПО MySQL под коммерческой лицензией

- При линковании программы с любым GPL кодом из ПО MySQL, в тех случаях, когда вы не хотите, чтобы готовый продукт подпадал под действие GPL (например, продукт разрабатывается как коммерческий или существуют какие-либо другие причины не открывать добавленный программный код, который не подпадает под действие GPL). При покупке коммерческой лицензии не используется ПО MySQL под лицензией GPL, даже несмотря на то, что это один и тот же код.
- В случае распространения приложения, не защищенного лицензией GPL, которое предназначено для работы исключительно с ПО MySQL и поставляется вместе с ним. Такой вариант решения в действительности считается связыванием, даже если оно осуществляется по сети.
- В случае, когда требуется распространять ПО MySQL без предоставления исходного кода, как того требует лицензия GPL.



# Администрирование БД

---

- Конфигурирование сервера MySQL осуществляется при помощи файлов параметров
- В Unix считывание принятых по умолчанию параметров MySQL производится из следующих файлов:
  - `/etc/my.cnf` Общие параметры
  - `DATADIR/my.cnf` Параметры для сервера
  - `defaults-extra-file` Файл, указанный при помощи `-defaults-extra-file=#`
  - `~/.my.cnf` Параметры для пользователя
  - `DATADIR` является каталогом данных MySQL (обычно `/usr/local/mysql/data` для бинарной установки или `/usr/local/var` для установки из исходных текстов).
- В Windows считывание принятых по умолчанию параметров MySQL производится из следующих файлов:
  - `windows-system-directory\my.ini` Общие параметры сервера
  - `C:\my.cnf` Общие параметры пользователя

# Установка нескольких серверов

- В некоторых случаях необходимо, чтобы на одном компьютере работало несколько различных серверов `mysqld`:
  - например, если нужно запустить для тестирования новую версию MySQL совместно со старой версией, которая находится в работе, или когда разным пользователям нужно предоставить доступ к различным серверам `mysqld`, с которыми они умеют обращаться самостоятельно.
- Один из способов запустить новый сервер -- указать ему другой канал и порт, например, таким образом:
  - `shell> MYSQL_UNIX_PORT=/tmp/mysqld-new.sock`
  - `shell> MYSQL_TCP_PORT=3307`
  - `shell> export MYSQL_UNIX_PORT MYSQL_TCP_PORT`
  - `shell> scripts/mysql_install_db`
  - `shell> bin/safe_mysqld &`



# Управление доступом

---

- MySQL идентифицирует пользователя как по имени хоста, так и по имени пользователя, т.к. нет оснований полагать что данное имя пользователя принадлежит во всей Сети единственному человеку.
- Например, пользователь joe, устанавливающий соединение с office.com, вовсе не обязательно один и тот же человек, что и пользователь joe, подключающийся с elsewhere.com.
- MySQL решает эту проблему, обеспечивая возможность различать пользователей, подключающихся с различных хостов под одним и тем же именем пользователя: вы можете предоставлять joe один набор привилегий, если он подключается с office.com, и другой набор привилегий, если joe подключается с elsewhere.com.
- Управление доступом в MySQL осуществляется в два этапа:
  - Этап 1: сервер проверяет, имеется ли у вас вообще разрешение на подключение.
  - Этап 2: если таковое имеется, сервер проверяет каждый из ваших запросов, чтобы убедиться в том, что у вас имеется достаточно привилегий для его выполнения.
- Например, если вы пытаетесь выбрать строки в таблице базы данных или удалить таблицу из базы данных, сервер в первом случае проверяет, имеется ли у вас для этой таблицы привилегия SELECT, а во втором - имеется ли у вас для этой базы данных привилегия DROP.
- На обеих стадиях управления доступом сервер использует таблицы user, db и host из базы данных mysql.



# Безопасность

---

- При работе в MySQL старайтесь следовать приведенным ниже инструкциям:
- **Не предоставляйте никому (за исключением пользователя mysql под именем root) доступа к таблице user в базе данных mysql! Это чрезвычайно важно. В MySQL зашифрованный пароль является реальным паролем.** Узнав пароль, занесенный в таблицу user, и имея доступ к удаленному компьютеру, занесенному в соответствующую учетную запись, **войти в систему под именем зарегистрированного владельца пароля легко может кто угодно.**
- Изучите систему прав доступа MySQL. Для управления доступом к MySQL служат команды GRANT и REVOKE. Предоставляйте ровно столько прав, сколько необходимо, и не больше. Никогда не предоставляйте права всем хостам. Полезно проводить следующие контрольные проверки:
  - Выполните команду `mysql -u root`. Если удастся успешно установить соединение с сервером без получения запроса пароля, значит, у вас имеются проблемы. Это означает, что кто угодно может подсоединиться к вашему серверу MySQL как клиент MySQL под именем root, получая таким образом право неограниченного доступа! Проанализируйте инструкцию по установке MySQL, обращая особое внимание на ту часть, которая касается задания пароля пользователя root.
  - С помощью команды `SHOW GRANTS` проверьте, кто и к каким ресурсам имеет доступ. Воспользуйтесь командой `REVOKE`, отмените права доступа, которые не являются необходимыми.
- Не храните в базе данных незашифрованных паролей. Если злоумышленнику удастся получить доступ на ваш компьютер, то в его руках окажется полный список паролей, которыми он может воспользоваться. Применяйте для шифрования MD5(), SHA1() или другие односторонние хеш-функции.



# Управление доступом, этап 1

---

- При попытке соединения с сервером MySQL он либо устанавливает соединение, либо отказывает в нем - на основе данных о вашей личности и того, можете ли вы подтвердить их соответствующим паролем. Если нет, сервер полностью отказывает вам в доступе. В противном случае сервер устанавливает соединение, затем переходит ко второму этапу и ожидает запросов.
- Личность задается двумя порциями информации:
  - хостом, с которого вы подсоединяетесь
  - вашим именем пользователя MySQL
- Проверка личности осуществляется с помощью трех полей контекста таблицы user (Host, User и Password). Сервер устанавливает соединение только в том случае, если находит в таблице user запись, в которой имя хоста и имя пользователя совпадают с введенными вами, и вы указываете правильный пароль.



# Управление доступом, этап 2: верификация запросов

- После установления соединения сервер приступает к выполнению второго этапа. Для каждого поступающего запроса сервер проверяет, имеется ли у вас достаточно привилегий для его выполнения, основываясь при этом на типе операции, которую вы хотите выполнить.
- Теперь в действие вступают поля привилегий в таблицах привилегий. Информация о привилегиях может находиться в любой из таблиц привилегий - `user`, `db`, `host`, `tables_priv` или `columns_priv`.
- Обработка таблиц привилегий осуществляется с помощью команд `GRANT` и `REVOKE`.
- Таблица `user` предоставляет привилегии глобального характера, и их применение не зависит от того, в какой базе вы работаете в данный момент.
  - Например, если таблица `user` предоставляет пользователю привилегию `DELETE`, он может удалять строки из любой базы данных на серверном хосте! Иными словами, привилегии в таблице `user` - это привилегии суперпользователя. Поэтому целесообразно предоставлять привилегии в таблице `user` только суперпользователям, таким как администраторы сервера или администраторы баз данных. Что касается других пользователей, для их привилегий в таблице `user` следует установить значение 'N' и предоставлять им привилегии только на уровне баз данных, используя для этого таблицы `db` и `host`.
- Таблицы `db` и `host` предоставляют привилегии на уровне базы данных.
- Таблицы `tables_priv` и `columns_priv` предоставляют привилегии соответственно на уровне таблиц и столбцов.

# Управление учетными записями пользователей MySQL

- Команды GRANT и REVOKE позволяют системным администраторам создавать пользователей MySQL, а также предоставлять права пользователям или лишать их прав на четырех уровнях привилегий:
- **Глобальный уровень**
  - Глобальные привилегии применяются ко всем базам данных на указанном сервере. Эти привилегии хранятся в таблице `mysql.user`.
- **Уровень базы данных**
  - Привилегии базы данных применяются ко всем таблицам указанной базы данных. Эти привилегии хранятся в таблицах `mysql.db` и `mysql.host`.
- **Уровень таблицы**
  - Привилегии таблицы применяются ко всем столбцам указанной таблицы. Эти привилегии хранятся в таблице `mysql.tables_priv`.
- **Уровень столбца**
  - Привилегии столбца применяются к отдельным столбцам указанной таблицы. Эти привилегии хранятся в таблице `mysql.columns_priv`.

# Использование безопасных соединений

- MySQL поддерживает шифрованные SSL-соединения.
- По умолчанию в MySQL используются незашифрованные соединения между клиентом и сервером. Это означает, что просматривать все данные, передаваемые между клиентом и сервером, может кто угодно. На практике можно даже изменять данные во время передачи их от клиента к серверу и наоборот. Помимо того, иногда возникает необходимость передать действительно секретные данные через общедоступную сеть - в таких случаях использование незашифрованных соединений просто неприемлемо.
- В протоколе SSL используются различные алгоритмы шифрования, обеспечивающие безопасность для данных, передаваемых через общедоступные сети. Этот протокол содержит средства, позволяющие обнаруживать любые изменения, потери и повторы данных. В протоколе SSL также применяются алгоритмы для проведения идентификации при помощи стандарта X509.

# Резервное копирование баз данных



- Поскольку таблицы MySQL хранятся в виде файлов, то резервное копирование выполняется легко. Чтобы резервная копия была согласованной, выполните на выбранных таблицах `LOCK TABLES`, а затем `FLUSH TABLES` для этих таблиц.
- При этом требуется блокировка только на чтение; поэтому другие потоки смогут продолжать запросы на таблицах в то время, пока будут создаваться копии файлов из каталога базы данных.
- Команда `FLUSH TABLE` обеспечивает гарантию того, что все активные индексные страницы будут записаны на диск прежде, чем начнется резервное копирование.