# Лабораторна робота 2

## Аналіз даних про стан роботи сервера та БД

## ХІД РОБОТИ

### Завдання 1. Отриманні інформації о базах данных и таблицах

Как быть, если вы забыли имя базы или таблицы, или структуру какой-либо из таблиц (например имена столбцов)? В MySQL эта проблема решается при помощи нескольких команд, выводящих информацию о базе данных и содержащихся в ней таблицах.

Вы уже познакомились с командой SHOW DATABASES, выводящей список управляемых сервером баз данных. Определить, какая из них выбрана в данный момент, можно с помощью функции DATABASE():

mysql> SELECT DATABASE();

+------------+

| DATABASE() |

+------------+

| menagerie |

+------------+

Если ни одна из баз не выбрана, результат будет пуст.

Выяснить, какие таблицы содержит текущая база данных (что необходимо, если, например, никак не получается вспомнить имя нужной таблицы), можно при помощи следующей команды:

mysql> SHOW TABLES;

+---------------------+

| Tables in menagerie |

+---------------------+

| event |

| pet |

+---------------------+

Узнать структуру таблицы можно при помощи команды DESCRIBE, которая выводит информацию о каждом из столбцов таблицы:

mysql> DESCRIBE pet;

+---------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+---------+-------------+------+-----+---------+-------+

| name | varchar(20) | YES | | NULL | |

| owner | varchar(20) | YES | | NULL | |

| species | varchar(20) | YES | | NULL | |

| sex | char(1) | YES | | NULL | |

| birth | date | YES | | NULL | |

| death | date | YES | | NULL | |

+---------+-------------+------+-----+---------+-------+

Field - имя столбца, Type - тип данных, к которому относится этот столбец, NULL указывает, может ли данный столбец содержать значения NULL, Key - является ли этот столбец индексным, и, наконец, Default указывает значение данного столбца по умолчанию.

Если для таблицы созданы индексы, информацию о них можно получить с помощью команды SHOW INDEX FROM tbl\_name.

Можно использовать два альтернативных синтаксиса - tbl\_name FROM db\_name и db\_name.tbl\_name. Приведенные ниже два оператора эквивалентны:

mysql> SHOW INDEX FROM mytable FROM mydb;

mysql> SHOW INDEX FROM mydb.mytable;

Команда SHOW INDEX выводит информацию по индексу в формате, подобном формату вывода запроса SQLStatistics в ODBC. Выводятся следующие столбцы:

|  |  |
| --- | --- |
| **Столбец** | **Значение** |
| Table | Имя таблицы. |
| Non\_unique | 0 если индекс не может содержать дублирующихся значений. |
| Key\_name | Имя индекса. |
| Seq\_in\_index | Порядковый номер столбца в индексе, начиная с 1. |
| Column\_name | Имя столбца. |
| Collation | Как столбцы отсортированы в индексе. В MySQL в данном столбце могут присутствовать значения 'A' (в порядке возрастания) или NULL (не отсортирован). |
| Cardinality | Количество уникальных значений в индексе. Обновляется путем запуска isamchk -a. |
| Sub\_part Количество индексированных символов, если столбец индексируется частично. Если проиндексирован весь ключ, то будет содержаться значение NULL. |  |
| Null | Содержит значение 'YES', если столбец может содержать NULL. |
| Index\_type Используемый метод индексирования. |  |
| Comment | Различные примечания. На данный момент в версиях MySQL < 4.0.2 выдается, является индекс FULLTEXT или нет. |

SHOW TABLE STATUS [FROM db\_name] [LIKE wild]

Команда SHOW TABLE STATUS (новшество версии 3.23) работает как SHOW STATUS, но предоставляет большое количество информации по каждой таблице. Приведенный ниже список также можно получить, используя команду mysqlshow --status db\_name. Выводятся следующие столбцы:

|  |  |
| --- | --- |
| **Столбец** | **Значение** |
| Name | Имя таблицы. |
| Type | Тип таблицы. See section [7 Типы таблиц MySQL](http://www.mysql.ru/docs/man/Table_types.html). |
| Row\_format | Формат хранения строк (Fixed, Dynamic, или Compressed). |
| Rows | Количество строк. |
| Avg\_row\_length | Средняя длина строки. |
| Data\_length | Размер файла данных. |
| Max\_data\_length | Максимальная длина файла данных. |
| Index\_length | Длина индексного файла. |
| Data\_free | Количество распределенных, но не используемых байтов. |
| Auto\_increment | Следующее значение автоинкремента. |
| Create\_time | Время создания таблицы. |
| Update\_time | Время последнего обновления файла данных. |
| Check\_time | Время последней проверки таблицы. |
| Create\_options | Дополнительные параметры, использовавшиеся для команды CREATE TABLE. |
| Comment | Примечания, внесенные при создании таблицы (или информация о причинах, почему MySQL не может получить доступ к данным в таблицах). |

В таблицах InnoDB информация о свободном дисковом пространстве в таблице заносится в ячейки примечаний к таблице.

Додатково

http://linux-notes.org/razmer-bazy-tablitsy-v-mysql/

### Завдання 2. Аналіз стану сервера

### SHOW STATUS

Команда SHOW STATUS предоставляет информацию по состоянию сервера (как mysqladmin extended-status). Пример выходных данных приведен ниже (формат и числа могут иметь некоторые отличия):

+--------------------------+------------+

| Variable\_name | Value |

+--------------------------+------------+

| Aborted\_clients | 0 |

| Aborted\_connects | 0 |

| Bytes\_received | 155372598 |

| Bytes\_sent | 1176560426 |

| Connections | 30023 |

| Created\_tmp\_disk\_tables | 0 |

| Created\_tmp\_tables | 8340 |

| Created\_tmp\_files | 60 |

| Delayed\_insert\_threads | 0 |

| Delayed\_writes | 0 |

| Delayed\_errors | 0 |

| Flush\_commands | 1 |

| Handler\_delete | 462604 |

| Handler\_read\_first | 105881 |

| Handler\_read\_key | 27820558 |

| Handler\_read\_next | 390681754 |

| Handler\_read\_prev | 6022500 |

| Handler\_read\_rnd | 30546748 |

| Handler\_read\_rnd\_next | 246216530 |

| Handler\_update | 16945404 |

| Handler\_write | 60356676 |

| Key\_blocks\_used | 14955 |

| Key\_read\_requests | 96854827 |

| Key\_reads | 162040 |

| Key\_write\_requests | 7589728 |

| Key\_writes | 3813196 |

| Max\_used\_connections | 0 |

| Not\_flushed\_key\_blocks | 0 |

| Not\_flushed\_delayed\_rows | 0 |

| Open\_tables | 1 |

| Open\_files | 2 |

| Open\_streams | 0 |

| Opened\_tables | 44600 |

| Questions | 2026873 |

| Select\_full\_join | 0 |

| Select\_full\_range\_join | 0 |

| Select\_range | 99646 |

| Select\_range\_check | 0 |

| Select\_scan | 30802 |

| Slave\_running | OFF |

| Slave\_open\_temp\_tables | 0 |

| Slow\_launch\_threads | 0 |

| Slow\_queries | 0 |

| Sort\_merge\_passes | 30 |

| Sort\_range | 500 |

| Sort\_rows | 30296250 |

| Sort\_scan | 4650 |

| Table\_locks\_immediate | 1920382 |

| Table\_locks\_waited | 0 |

| Threads\_cached | 0 |

| Threads\_created | 30022 |

| Threads\_connected | 1 |

| Threads\_running | 1 |

| Uptime | 80380 |

+--------------------------+------------+

Приведенные выше переменные состояния имеют следующие значения:

|  |  |
| --- | --- |
| **Переменная** | **Значение** |
| Aborted\_clients | Количество соединений, отмененных по причине отключения клиента без надлежащего закрытия соединения. See section [A.2.9 Коммуникационные ошибки / Оборванные соединения](http://www.mysql.ru/docs/man/Communication_errors.html). |
| Aborted\_connects | Количество неудачных попыток подсоединения к серверу MySQL. See section [A.2.9 Коммуникационные ошибки / Оборванные соединения](http://www.mysql.ru/docs/man/Communication_errors.html). |
| Bytes\_received | Количество байтов, полученных от всех клиентов. |
| Bytes\_sent | Количество байтов, отправленных всем клиентам. |
| Com\_xxx | Количество запусков каждой команды xxx. |
| Connections | Количество попыток подсоединения к серверу MySQL. |
| Created\_tmp\_tables | Количество неявных временных таблиц на диске, созданных во время выполнения операторов. |
| Created\_tmp\_tables | Количество неявных временных таблиц в памяти, созданных во время выполнения операторов. |
| Created\_tmp\_files | Количество созданных временных файлов mysqld. |
| Delayed\_insert\_threads | Количество используемых потоков вставки данных в режиме insert delayed. |
| Delayed\_writes | Количество строк, вставленных при помощи команды INSERT DELAYED. |
| Delayed\_errors | Количество записанных при помощи команды INSERT DELAYED строк, в которых произошли какие-либо ошибки (возможно, duplicate key). |
| Flush\_commands | Количество запущенных команд FLUSH. |
| Handler\_commit | Количество внутренних команд COMMIT. |
| Handler\_delete | Количество удалений строки из таблицы. |
| Handler\_read\_first | Количество считываний из индекса первой записи. Если это значение высокое, то, по всей вероятности, сервер осуществляет много полных индексных сканирований, например, SELECT col1 FROM foo, предполагая, что col1проиндексирован. |
| Handler\_read\_key | Количество запросов на чтение строки, основанных на ключе. Высокое значение переменной говорит о том, что ваши запросы и таблицы проиндексированы надлежащим образом. |
| Handler\_read\_next | Количество запросов на чтение следующей строки в порядке расположения ключей. Это значение будет увеличиваться, если производится запрос индексного столбца с ограничением по размеру. Значение также увеличивается во время проведения индексного сканирования. |
| Handler\_read\_prev | Количество запросов на чтение предыдущей строки в порядке расположения ключей. В большинстве случаев используется для оптимизации ORDER BY ... DESC. |
| Handler\_read\_rnd | Количество запросов на чтение строки, основанных на фиксированной позиции. Значение будет высоким, если выполняется много запросов, требующих сортировки результатов. |
| Handler\_read\_rnd\_next | Количество запросов на чтение следующей строки из файла данных. Данное значение будет высоким, если производится много сканирований таблиц. Обычно это означает, что ваши таблицы не проиндексированы надлежащим образом или ваши запросы не используют преимущества индексов. |
| Handler\_rollback | Количество внутренних команд ROLLBACK. |
| Handler\_update | Количество запросов на обновление строки в таблице. |
| Handler\_write | Количество запросов на вставку строки в таблицу. |
| Key\_blocks\_used | Количество используемых блоков в кэше ключей. |
| Key\_read\_requests | Количество запросов на чтение блока ключей из кэша. |
| Key\_reads | Количество физических считываний блока ключей с диска. |
| Key\_write\_requests | Количество запросов на запись блока ключей в кэш. |
| Key\_writes | Количество физических записей блоков ключей на диск. |
| Max\_used\_connections | Максимальное количество одновременно используемых соединений. |
| Not\_flushed\_key\_blocks | Блоки ключей в кэше ключей, которые были изменены, но еще не записаны на диск. |
| Not\_flushed\_delayed\_rows | Количество строк, стоящих в очереди на запись в запросах INSERT DELAY. |
| Open\_tables | Количество открытых таблиц. |
| Open\_files | Количество открытых файлов. |
| Open\_streams | Количество открытых потоков (в основном используется для журналирования). |
| Opened\_tables | Количество открывавшихся таблиц. |
| Rpl\_status | Статус отказобезопасной репликации (еще не используется). |
| Select\_full\_join | Количество соединений без ключей (если это значение равно 0, необходимо внимательно проверить индексы своих таблиц). |
| Select\_full\_range\_join | Количество соединений, где был использован поиск по диапазону в справочной таблице. |
| Select\_range | Количество соединений, в которых использовались диапазоны в первой таблице. (Обычно это значение не критично, даже если оно велико) |
| Select\_scan | Количество соединений, в которых проводилось первое сканирование первой таблицы. |
| Select\_range\_check | Количество соединений без ключей, в которых проверка использования ключей производится после каждой строки (если это значение равно 0, необходимо внимательно проверить индексы своих таблиц). |
| Questions | Количество запросов, направленных на сервер. |
| Slave\_open\_temp\_tables | Количество временных таблиц, открытых в настоящий момент потоком подчиненного компьютера. |
| Slave\_running | Содержит значение ON, если это подчиненный компьютер, подключенный к головному компьютеру. |
| Slow\_launch\_threads | Количество потоков, создание которых заняло больше, чем указано в slow\_launch\_time. |
| Slow\_queries | Количество запросов, обработка которых заняла больше времени, чем long\_query\_time. See section [4.9.5 Журнал медленных запросов](http://www.mysql.ru/docs/man/Slow_query_log.html). |
| Sort\_merge\_passes | Количество объединений, осуществленных алгоритмом сортировки. Если это значение велико, следует увеличить sort\_buffer. |
| Sort\_range | Количество сортировок, которые осуществлялись в диапазонах. |
| Sort\_rows | Количество отсортированных строк. |
| Sort\_scan | Количество сортировок, осуществленных путем сканирования таблицы. |
| ssl\_xxx | Переменные, используемые SSL; еще не реализовано. |
| Table\_locks\_immediate | Количество запросов на немедленную блокировку таблицы. Доступно начиная с версии 3.23.33. |
| Table\_locks\_waited | Количество запросов, когда немедленная блокировка не могла быть осуществлена и требовалось время на ожидание. Если это значение велико, и у вас есть проблемы с производительностью, сначала необходимо оптимизировать свои запросы, а затем либо разделить таблицы, либо использовать репликацию. Доступно начиная с версии 3.23.33. |
| Threads\_cached | Количество потоков в кэше потоков. |
| Threads\_connected | Количество открытых в настоящий момент соединений. |
| Threads\_created | Количество потоков, созданных для управления соединениями. |
| Threads\_running | Количество не простаивающих потоков. |
| Uptime | Время в секундах, в течение которого сервер находится в работе. |

Некоторые примечания к приведенной выше информации:

* Если значение Opened\_tables велико, возможно, что значение переменной table\_cache слишком мало.
* Если значение Key\_reads велико, возможно, что значение переменной key\_buffer\_size слишком мало. Частоту успешных обращений к кэшу можно вычислить при помощи Key\_reads/Key\_read\_requests.
* Если значение Handler\_read\_rnd велико, возможно, поступает слишком много запросов, требующих от MySQL полного сканирования таблиц или у вас есть соединения, которые не используют ключи надлежащим образом.
* Если значение Threads\_created велико, возможно, необходимо увеличить значение переменной thread\_cache\_size. Частоту успешных обращений к кэшу можно вычислить при помощи Threads\_created/Connections.
* Если значение Created\_tmp\_disk\_tables велико, возможно, необходимо увеличить значение переменной tmp\_table\_size, чтобы временные таблицы располагались в памяти, а не на жестком диске.

**Завдання 2. Отримання інформації про стан системних змінних**

Команда SHOW VARIABLES отображает значения некоторых системных переменных MySQL. Такую же информацию можно получить и при помощи команды mysqladmin variables. Если установленные по умолчанию значения вам не подходят, большинство из этих переменных можно задать, используя параметры командной строки при запуске mysqld. See section [4.1.1 Параметры командной строки mysqld](http://www.mysql.ru/docs/man/Command-line_options.html).

Ниже приведен пример выходных данных; формат и числа могут иметь некоторые отличия:

+------------------------------+---------------------------+

| Variable\_name | Value |

+------------------------------+---------------------------+

| ansi\_mode | OFF |

| back\_log | 50 |

| basedir | /my/monty/ |

| bdb\_cache\_size | 16777216 |

| bdb\_log\_buffer\_size | 32768 |

| bdb\_home | /my/monty/data/ |

| bdb\_max\_lock | 10000 |

| bdb\_logdir | |

| bdb\_shared\_data | OFF |

| bdb\_tmpdir | /tmp/ |

| binlog\_cache\_size | 32768 |

| concurrent\_insert | ON |

| connect\_timeout | 5 |

| datadir | /my/monty/data/ |

| delay\_key\_write | ON |

| delayed\_insert\_limit | 100 |

| delayed\_insert\_timeout | 300 |

| delayed\_queue\_size | 1000 |

| flush | OFF |

| flush\_time | 0 |

| ft\_min\_word\_len | 4 |

| ft\_max\_word\_len | 254 |

| ft\_max\_word\_len\_for\_sort | 20 |

| ft\_boolean\_syntax | + -><()~\* |

| have\_bdb | YES |

| have\_innodb | YES |

| have\_raid | YES |

| have\_openssl | NO |

| init\_file | |

| interactive\_timeout | 28800 |

| join\_buffer\_size | 131072 |

| key\_buffer\_size | 16776192 |

| language | /my/monty/share/english/ |

| large\_files\_support | ON |

| log | OFF |

| log\_update | OFF |

| log\_bin | OFF |

| log\_slave\_updates | OFF |

| long\_query\_time | 10 |

| low\_priority\_updates | OFF |

| lower\_case\_table\_names | 0 |

| max\_allowed\_packet | 1048576 |

| max\_binlog\_cache\_size | 4294967295 |

| max\_connections | 100 |

| max\_connect\_errors | 10 |

| max\_delayed\_threads | 20 |

| max\_heap\_table\_size | 16777216 |

| max\_join\_size | 4294967295 |

| max\_sort\_length | 1024 |

| max\_tmp\_tables | 32 |

| max\_write\_lock\_count | 4294967295 |

| myisam\_bulk\_insert\_tree\_size | 8388608 |

| myisam\_recover\_options | DEFAULT |

| myisam\_sort\_buffer\_size | 8388608 |

| net\_buffer\_length | 16384 |

| net\_read\_timeout | 30 |

| net\_retry\_count | 10 |

| net\_write\_timeout | 60 |

| open\_files\_limit | 0 |

| pid\_file | /my/monty/data/donna.pid |

| port | 3306 |

| protocol\_version | 10 |

| record\_buffer | 131072 |

| query\_buffer\_size | 0 |

| query\_cache\_limit | 1048576 |

| query\_cache\_size | 16768060 |

| query\_cache\_startup\_type | 1 |

| safe\_show\_database | OFF |

| server\_id | 0 |

| skip\_locking | ON |

| skip\_networking | OFF |

| skip\_show\_database | OFF |

| slow\_launch\_time | 2 |

| socket | /tmp/mysql.sock |

| sort\_buffer | 2097116 |

| table\_cache | 64 |

| table\_type | MYISAM |

| thread\_cache\_size | 4 |

| thread\_stack | 65536 |

| tmp\_table\_size | 1048576 |

| tmpdir | /tmp/ |

| version | 3.23.29a-gamma-debug |

| wait\_timeout | 28800 |

+------------------------------+---------------------------+

Ниже описаны все параметры. Значения размеров буферов, длины и размеры стеков приведены в байтах. Значения можно указать с суффиксом 'K' или 'M', чтобы обозначить килобайты или мегабайты. Например, 16M означает 16 мегабайт. Регистр буквы суффикса не имеет значения, 16M и 16m обозначают одно и то же:

* ansi\_mode. Имеет значение ON, если mysqld был запущен с параметром --ansi. See section [1.9.2 Запуск MySQL в режиме ANSI](http://www.mysql.ru/docs/man/ANSI_mode.html).
* back\_log Количество поддерживаемых MySQL запросов на соединение, находящихся в очереди. Этот параметр начинает играть роль, когда главный поток MySQL получает очень много запросов на соединение за короткий промежуток времени. У главного потока уходит некоторое время (хотя очень небольшое) на проверку соединения и запуск нового потока. Значение back\_log показывает, сколько запросов может находиться в очереди на этом коротком промежутке времени, прежде чем MySQL прекратит отвечать на новые запросы. Данное значение необходимо увеличить только в том случае, если ожидается большое количество соединений на протяжении короткого промежутка времени. Иначе говоря, это значение является размером очереди ожидания входящих соединений TCP/IP. У вашей операционной системы есть свое собственное ограничение на размер этой очереди. За более подробной информацией обращайтесь на страницу руководства ОС Unix по системному вызову listen(2). Чтобы узнать максимальное значение для этой переменной, обратитесь к документации по своей операционной системе. Попытка установить значение back\_log выше, чем допускается в вашей операционной системе, не принесет положительного результата.
* basedir Значение параметра --basedir.
* bdb\_cache\_size Буфер, выделенный для индекса кэша и строк таблиц BDB. Если таблицы BDB не используются, необходимо запустить mysqld с параметром --skip-bdb, чтобы не расходовать память на этот кэш.
* bdb\_log\_buffer\_size Буфер, выделенный для индекса кэша и строк таблиц BDB. Если таблицы BDB не используются, его значение необходимо установить в 0 или запустить mysqld с параметром --skip-bdb, чтобы не расходовать память на этот кэш.
* bdb\_home Значение параметра --bdb-home.
* bdb\_max\_lock Максимальное количество блокировок (по умолчанию 1000), которые можно установить на таблицу BDB. Этот параметр необходимо увеличить, если возникают ошибки типа bdb: Lock table is out of available locks или Got error 12 from ... при выполнении длинных транзакций или когда mysqld должен просмотреть много строк для вычисления запроса.
* bdb\_logdir Значение параметра --bdb-logdir.
* bdb\_shared\_data Содержит значение ON, если используется параметр --bdb-shared-data.
* bdb\_tmpdir Значение параметра --bdb-tmpdir.
* binlog\_cache\_size. Размер кэша для хранения операторов бинарного журнала SQL во время транзакции. Если часто используются большие транзакции со значительным количеством операторов, то в целях повышения производительности значение этого параметра можно увеличить. See section [6.7.1 Синтаксис команд BEGIN/COMMIT/ROLLBACK](http://www.mysql.ru/docs/man/COMMIT.html).
* character\_set Принятый по умолчанию набор символов.
* character\_sets Поддерживаемые наборы символов.
* concurrent\_inserts Если установлено значение ON (принятое по умолчанию), MySQL обеспечивает возможность использовать команду INSERTна таблицах MyISAM одновременно с выполнением над этими же таблицами запросов SELECT. Этот параметр можно отключить, запустив mysqldс параметром --safe или --skip-new.
* connect\_timeout Количество времени в секундах, на протяжении которого сервер mysqld ожидает поступления пакета соединения, после чего генерирует Bad handshake.
* datadir Значение параметра --datadir.
* delay\_key\_write Если включен (по умолчанию - включен), MySQL будет учитывать параметр DELAY\_KEY\_WRITE для команды CREATE TABLE. Это означает, что ключевой буфер таблиц с данным параметром будет сбрасываться на диск не при каждом обновлении индексов, а только при закрытии таблицы. Такой режим работы значительно ускоряет запись ключей, однако в случае использования данного параметра необходимо будет также добавить автоматическую проверку всех таблиц при помощи myisamchk --fast --force. Обратите внимание: если запустить mysqld с параметром --delay-key-write-for-all-tables, то все таблицы будут обрабатываться таким образом, как будто они были созданы с применением параметра delay\_key\_write. Этот флаг можно снять, запустив mysqld с параметром --skip-new или --safe-mode.
* delayed\_insert\_limit После вставки строк delayed\_insert\_limit обработчик INSERT DELAYED проверит, остались ли незавершенные операторы SELECT. Если да, то перед тем, как приступить к выполнению следующих действий, они будут выполнены.
* delayed\_insert\_timeout Временной промежуток, в течение которого процесс INSERT DELAYED должен ожидать операторов INSERT до прекращения выполнения задачи.
* delayed\_queue\_size Размер очереди (в строках), который должен быть назначен для обработки команды INSERT DELAYED. При переполнении очереди все клиенты, выполняющие команду INSERT DELAYED, будут ожидать, пока в очереди снова появится свободное место.
* flush Значение этой переменной будет ON, если вы запустили MySQL с параметром --flush.
* flush\_time Если значение этой переменной отличается от нуля, то каждые flush\_time секунд все таблицы будут закрываться (чтобы освободить ресурсы и записать информацию на диск). Мы рекомендуем использовать этот параметр только для Windows 9x/Me или на системах с ограниченным количеством ресурсов.
* ft\_min\_word\_len Минимальная длина слова, включаемого в индекс FULLTEXT. **Примечание: индексы FULLTEXT после изменения этой переменной должны быть скомпонованы заново (это новый параметр в MySQL 4.0)**.
* ft\_max\_word\_len Максимальная длина слова, включаемого в индекс FULLTEXT. **Примечание: индексы FULLTEXT после изменения этой переменной должны быть скомпонованы заново (это новый параметр в MySQL 4.0)**.
* ft\_max\_word\_len\_sort Максимальная длина слова в индексе FULLTEXT; эта длина будет использоваться для метода быстрого восстановления индекса в командах REPAIR, CREATE INDEX или ALTER TABLE. Более длинные слова вставляются дольше. При увеличении значения ft\_max\_word\_len\_sort MySQL будет создавать временные файлы большего размера (таким образом будет замедляться работа из-за возрастания нагрузки на порт ввода/вывода диска), а также будет помещать меньше ключей в один блок сортировки (что, опять же, способствует снижению производительности). Если значение переменной ft\_max\_word\_len\_sort слишком мало, MySQL будет вставлять большое количество слов достаточно медленно, но короткие слова будут вставляться очень быстро.
* ft\_boolean\_syntax Список операторов, поддерживаемых командой MATCH ... AGAINST(... IN BOOLEAN MODE). See section [6.8 Полнотекстовый поиск в MySQL](http://www.mysql.ru/docs/man/Fulltext_Search.html).
* have\_innodb Значение YES, если mysqld поддерживает таблицы InnoDB. Значение DISABLED, если используется параметр --skip-innodb.
* have\_bdb Значение YES, если mysqld поддерживает таблицы Berkeley DB. Значение DISABLED, если используется параметр --skip-bdb.
* have\_raid Значение YES, если mysqld поддерживает параметр RAID.
* have\_openssl Значение YES, если mysqld поддерживает SSL (шифрование) в протоколе клиент/сервер.
* init\_file Имя файла, указанного при помощи параметра --init-file при запуске сервера. Это файл операторов SQL, которые сервер должен выполнить при запуске.
* interactive\_timeout Количество времени в секундах, на протяжении которого сервер ожидает активности со стороны интерактивного соединения, прежде чем закрыть его. Интерактивный клиент - это клиент, который использует параметр CLIENT\_INTERACTIVE дляmysql\_real\_connect(). См. также информацию по wait\_timeout.
* join\_buffer\_size Размер буфера, используемого для полных объединений (объединения, в которых не применяются индексы). Буфер выделяется один раз для каждого полного объединения двух таблиц. Если нужно ускорить полное объединение и невозможно добавить индексы, следует увеличить это значение (обычно добавление индексов является лучшим способом добиться быстрых объединений).
* key\_buffer\_size Блоки индексов буферизированы и доступ к ним разрешен всем потокам. key\_buffer\_size - размер буфера, используемого для блоков индексов. Чтобы улучшить обработку индексов (для всех операций чтения и записи нескольких элементов), необходимо увеличить это значение настолько, насколько возможно. Размер 64 Мб для компьютера с 256 Мб (именно такой компьютер в основном используется для работы с MySQL) - довольно распространенная конфигурация. Тем не менее, если задать слишком большое значение (например, больше 50% от общего количества памяти), система может начать сохранять временные файлы на диске, что значительно снизит производительность. Помните, что, поскольку MySQL не производит кэширования считываемых данных, нужно оставлять определенное пространство для кэша файловой системы операционной системы. Производительность буфера ключей можно проверить, выполнив команду show status и проверив значения переменных Key\_read\_requests, Key\_reads, Key\_write\_requests и Key\_writes. Отношение значений Key\_reads/Key\_read\_request обычно должно быть < 0,01. Отношение Key\_write/Key\_write\_requests примерно равно 1, если в основном используются обновления/удаления, но может быть и меньше, если чаще используются обновления сразу многих элементов или еслиdelay\_key\_write. See section [4.5.6 Синтаксис команды SHOW](http://www.mysql.ru/docs/man/SHOW.html). Чтобы еще больше увеличить скорость при одновременной записи большого количества строк, используйте команду LOCK TABLES. See section [6.7.2 Синтаксис команд LOCK TABLES/UNLOCK TABLES](http://www.mysql.ru/docs/man/LOCK_TABLES.html).
* language Язык, используемый для вывода сообщений об ошибках.
* large\_file\_support Если mysqld был откомпилирован с параметрами для поддержки больших файлов.
* locked\_in\_memory Если mysqld был заблокирован в памяти при помощи --memlock
* log Если включено занесение в журнал всех запросов.
* log\_update Если включен журнал обновлений.
* log\_bin Если включен бинарный журнал.
* log\_slave\_updates Если обновления от подчиненного компьютера должны заноситься в журнал.
* long\_query\_time Если обработка запроса отнимает больше указанного промежутка времени (в секундах), значение счетчика Slow\_queriesбудет увеличено. Если используется параметр --log-slow-queries, запрос будет записан в журнал медленных запросов. See section [4.9.5 Журнал медленных запросов](http://www.mysql.ru/docs/man/Slow_query_log.html).
* lower\_case\_table\_names Если установлено значение 1, имена таблиц будут сохранятся на диск с использованием строчных букв, а имена таблиц не будут чувствительны к регистру. See section [6.1.3 Чувствительность имен к регистру](http://www.mysql.ru/docs/man/Name_case_sensitivity.html).
* max\_allowed\_packet Максимальный размер одного пакета. Изначально размер буфера сообщений устанавливается в net\_buffer\_length байтов, но при необходимости может возрасти до max\_allowed\_packet байтов. Это значение по умолчанию не настолько велико, чтобы отсеивать большие (возможно ошибочные) пакеты. Если используются большие столбцы BLOB, его необходимо увеличить. Значение должно быть не меньше самого большого BLOB, который будет использоваться. Ограничение протокола для max\_allowed\_packet составляет 16 Мб в MySQL 3.23 и 2Гб в MySQL 4.0.
* max\_binlog\_cache\_size Если для транзакции с большим количеством операторов потребуется большее количество памяти, чем указано этим параметром, будет выдано сообщение об ошибке "Multi-statement transaction required more than 'max\_binlog\_cache\_size' bytes of storage" (для транзакции с большим количеством операторов требуется больше, чем 'max\_binlog\_cache\_size' байтов для хранения).
* max\_binlog\_size Доступно после версии 3.23.33. Если запись в бинарный журнал (репликация) превысит данное значение, производится ротация журнала. Нельзя установить значение, меньшее, чем 1024 байта или большее, чем 1 Гб. По умолчанию установлено значение 1 Гб.
* max\_connections Разрешенное количество одновременно подсоединенных клиентов. Увеличение этого значения увеличивает количество дескрипторов файлов, необходимых для mysqld. Ниже приведены примечания по ограничениям для дескрипторов. See section [A.2.5 Ошибка Too many connections](http://www.mysql.ru/docs/man/Too_many_connections.html).
* max\_connect\_errors Если количество прерванных соединений с удаленным компьютером превышает это число, дальнейшие попытки удаленного компьютера установить соединение будут заблокированы. Блокировку с удаленного компьютера можно снять при помощи команды FLUSH HOSTS.
* max\_delayed\_threads Не запускайте больше потоков, чем указано здесь, для обработки операторов INSERT DELAYED. Если попытаться вставить данные в новую таблицу, когда все потоки INSERT DELAYED заняты, строка будет вставлена таким образом, как будто атрибут DELAYED не был указан.
* max\_heap\_table\_size Не позволяет создавать динамические таблицы, размер которых превышает указанное значение.
* max\_join\_size Объединения, которые потенциально могут считывать более max\_join\_size записей, будут возвращать ошибку. Это значение нужно задавать, если ваши пользователи осуществляют объединения, которым недостает оператора WHERE, - такие объединения занимают много времени, а затем возвращают миллионы строк.
* max\_sort\_length Параметр определяет, сколько байтов следует использовать при сортировке значений BLOB или TEXT (обрабатываются только первые max\_sort\_length байтов каждого значения, остальные игнорируются).
* max\_user\_connections Максимальное количество активных соединений для одного пользователя (0 = без ограничений).
* max\_tmp\_tables (Этот параметр пока ни на что не влияет.) Максимальное количество временных таблиц, которые клиент может открывать одновременно.
* max\_write\_lock\_count После данного количества блокирования записей разрешается выполнить между ними несколько блокировок чтения.
* myisam\_bulk\_insert\_tree\_size В MySQL используется специальный древовидный кэш, чтобы быстрее производить групповые вставки (например INSERT ... SELECT, INSERT ... VALUES (...), (...), ... и LOAD DATA INFILE). Данная переменная ограничивает размер дерева кэша в байтах на поток. Если установить значение в 0, то оптимизация будет отключена. **Примечание**: этот кэш используется только для добавления данных к таблицам, которые уже содержат определенную информацию. По умолчанию установлено значение 8 Мб.
* myisam\_recover\_options Значение параметра --myisam-recover.
* myisam\_sort\_buffer\_size Буфер, который выделяется для сортировки индексов при выполнении команды REPAIR или для создания индексов при помощи команд CREATE INDEX или ALTER TABLE.
* myisam\_max\_extra\_sort\_file\_size. Если размер временного файла, используемого для быстрого создания индексов, превышает на указанный здесь объем используемый кэш ключей, то лучше отдать предпочтение методу кэша ключей. Такой метод применяется для того, чтобы для больших ключей символов в больших таблицах использовался более медленный метод кэширования ключей при создании таблиц. **Обратите внимание**: значение этого параметра задается в мегабайтах!
* myisam\_max\_sort\_file\_size Максимальный размер временного файла, который в MySQL может использоваться при восстановлении индекса (во время работы команд REPAIR, ALTER TABLE или LOAD DATA INFILE). Для файлов, размер которых превышает указанное значение, индекс будет создаваться при помощи кэша ключей (такой алгоритм работает несколько медленнее). **Обратите внимание**: значение этого параметра задается в мегабайтах!
* net\_buffer\_length В данное значение устанавливается в промежутках между запросами буфер соединения. Обычно это значение не изменяется, но если у вас очень мало памяти, можно установить его по размеру ожидаемого запроса (т.е. равным предполагаемой длине операторов SQL, отправляемых клиентами; если оператор превысит указанную длину, буфер будет автоматически увеличен как максимум до max\_allowed\_packet байтов).
* net\_read\_timeout Количество времени в секундах, на протяжении которого ожидаются дополнительные данные от соединения, пока не будет отменено чтение. Обратите внимание, что мы не ожидаем поступления данных от соединения, время ожидания определяется по write\_timeout. Также см. slave\_net\_timeout.
* net\_retry\_count Если чтение из порта связи было прервано, будет предпринято указанное количество попыток повторного чтения. Это значение должно быть достаточно высоким на FreeBSD, так как внутренние прерывания направляются на все потоки.
* net\_write\_timeout Время ожидания записи блока через соединение, пока запись не будет прервана (в секундах).
* open\_files\_limit Если это значение отлично от 0, то mysqld будет применять его для резервных дескрипторов файлов, используемых сsetrlimit(). Если это значение равно 0, то mysqld будет резервировать max\_connections\*5 или max\_connections + table\_cache\*2 (в зависимости от того, какое число больше) файлов. Если mysqld выдает ошибку 'Too many open files' (слишком много открытых файлов), данное значение необходимо увеличить.
* pid\_file Значение параметра --pid-file.
* port Значение параметра --port.
* protocol\_version Версия протокола, используемого сервером MySQL.
* record\_buffer Каждый поток, осуществляющий последовательное сканирование, выделяет буфер указанного размера для каждой сканируемой таблицы. Если проводится много последовательных сканирований, это значение можно увеличить.
* record\_rnd\_buffer При считывании строк после проведения сортировки в отсортированном порядке строки считываются через буфер, чтобы избежать операций поиска по диску. Если этот параметр не задан, то используется значение record\_buffer.
* query\_buffer\_size Изначальная величина буфера запросов. Если большинство запросов большого размера (например, при вставке записей сBLOB), его необходимо увеличить!
* query\_cache\_limit Результаты, превышающие это значение, не кэшируются (по умолчанию - 1Мб).
* query\_cache\_size Память, выделенная для хранения результатов старых запросов. Если значение установлено в 0, кэш запросов отключен (принято по умолчанию).
* query\_cache\_startup\_type Могут быть заданы следующие значения (только числовые):

|  |  |  |
| --- | --- | --- |
| **Значение** | **Объяснение** | **Примечание** |
| 0 | OFF | Не кэширует и не извлекает результаты. |
| 1 | ON | Кэширует все результаты, кроме запросов SELECT SQL\_NO\_CACHE .... |
| 2 | DEMAND | Кэширует только запросы SELECT SQL\_CACHE .... |

* safe\_show\_database Не отображаются базы данных, для которых у пользователя нет каких-либо привилегий по базам данных или таблицам. Таким образом можно увеличить степень безопасности, если вы не хотите предоставлять посторонним лицам возможность получать информацию о том, какие базы данных есть у других пользователей. Также см. skip\_show\_database.
* server\_id Значение параметра --server-id.
* skip\_locking Установлено в значение OFF, если mysqld использует внешнюю блокировку.
* skip\_networking Установлено в значение ON, если разрешаются только локальные соединения (через сокет).
* skip\_show\_database Эта переменная не позволяет выполнять команду SHOW DATABASES, если у пользователя нет привилегии PROCESS. Таким образом можно увеличить степень безопасности, если вы не хотите предоставлять посторонним лицам возможность получать информацию о том, какие базы данных есть у других пользователей. Также см. safe\_show\_database.
* slave\_net\_timeout Количество времени в секундах, в течение которого ожидаются дополнительные данные от подсоединенного головного или подчиненного компьютера прежде, чем будет отменено чтение.
* slow\_launch\_time Если создание потока занимает больше, чем указанное значение (в секундах), значение счетчика Slow\_launch\_threads будет увеличено.
* socket Сокет Unix, используемый сервером.
* sort\_buffer Каждый поток, которому необходимо произвести сортировку, выделяет буфер данного размера. Увеличение данного значения позволит ускорить выполнение операторов ORDER BY или GROUP BY. See section [A.4.4 Где MySQL хранит временные файлы](http://www.mysql.ru/docs/man/Temporary_files.html).
* table\_cache Количество открытых таблиц для всех потоков. С увеличением этого значения увеличивается количество дескрипторов файлов, необходимых для mysqld. Чтобы узнать, необходимо ли изменять значение кэша таблиц, следует проверить значение переменной Opened\_tables. См. раздел See section [A.4.4 Где MySQL хранит временные файлы](http://www.mysql.ru/docs/man/Temporary_files.html). Если у этой переменной большое значение, а команда FLUSH TABLES (которая закрывает все таблицы, а потом открывает их повторно) используется не часто, то необходимо увеличить ее значение. Чтобы получить дополнительную информацию по кэшу таблиц, см. раздел See section [5.4.7 Открытие и закрытие таблиц в MySQL](http://www.mysql.ru/docs/man/Table_cache.html).
* table\_type Принятый по умолчанию тип таблиц.
* thread\_cache\_size Определяет, сколько потоков должно сохраняться в кэше для повторного использования. После отключения клиента потоки клиента помещаются в кэш, если там не больше потоков, чем thread\_cache\_size. Все новые потоки сначала берутся из кэша, и только когда кэш становится пустым, создаются новые потоки. Значение этой переменной можно увеличить, чтобы повысить производительность, если создается много новых соединений (если потоки у вас хорошо организованы, обычно заметного улучшения производительности не наблюдается). Насколько эффективен текущий кэш потоков, можно определить по разнице между Connections и Threads\_created.
* thread\_concurrency В системе Solaris mysqld вызывает функцию thr\_setconcurrency() с этим значением. thr\_setconcurrency() позволяет приложению дать системе потоков подсказку по необходимому количеству потоков, которые должны быть запущены в одно и то же время.
* thread\_stack Размер стека для каждого потока. От данного значения зависит большое количество ограничений, обнаруживаемых при помощи теста crash-me. По умолчанию этот размер достаточен для нормальной работы. See section [5.1.4 Набор тестов MySQL (The MySQL Benchmark Suite)](http://www.mysql.ru/docs/man/MySQL_Benchmarks.html).
* timezone Часовой пояс, в котором находится сервер.
* tmp\_table\_size Если находящаяся в памяти временная таблица превысит этот размер, MySQL автоматически преобразует ее в дисковую таблицу MyISAM. Значение tmp\_table\_size следует увеличить, если выполняется много расширенных запросов GROUP BY и в вашей системе установлено значительное количество оперативной памяти.
* tmpdir Каталог, используемый для временных файлов и временных таблиц.
* version Номер версии сервера.
* wait\_timeout Время в секундах, на протяжении которого сервер ожидает активности соединения прежде, чем закрыть его. См. такжеinteractive\_timeout.

**Завдання 3. Отримання інформації про запущені потоки**

## SHOW PROCESSLIST

Команда SHOW [FULL] PROCESSLIST показывает, какие потоки запущены в настоящий момент. Эту информацию также можно получить при помощи команды mysqladmin processlist. Если у вас привилегия SUPER, можно просматривать все потоки, в противном случае - только свои потоки. See section [4.5.5 Синтаксис команды KILL](http://www.mysql.ru/docs/man/KILL.html). Если не используется параметр FULL, будут показаны только первые 100 символов каждого запроса.

Эта команда очень полезна, если выдается сообщение об ошибке 'too many connections' (слишком много соединений) и необходимо выяснить, что происходит. MySQL резервирует одно дополнительное соединение для клиента с привилегией SUPER, чтобы у вас всегда была возможность войти в систему и произвести проверку (предполагается, что вы не станете раздавать эту привилегию всем своим пользователям).

Некоторые состояния обычно можно увидеть в mysqladmin processlist.

* Checking table Поток осуществляет [автоматическую] проверку таблицы.
* Closing tables Означает, что поток записывает измененные данные таблиц на диск и закрывает использующиеся таблицы. Выполнение этой операции должно произойти быстро. Если на нее уходит значительное время, убедитесь, что диск не переполнен или что диск не используется слишком интенсивно.
* Connect Out Подчиненный компьютер, подсоединенный к головному компьютеру.
* Copying to tmp table on disk Набор временных результатов превысил tmp\_table\_size, и теперь поток изменяет таблицу временных данных, расположенную в памяти, на дисковую таблицу, чтобы сохранить память.
* Creating tmp table Поток создает временную таблицу, чтобы хранить часть результатов для запроса.
* deleting from main table При запуске первой части удаления нескольких таблиц удаление производится только начиная с первой таблицы.
* deleting from reference tables При запуске второй части удаления нескольких таблиц удаляются совпадающие строки из других таблиц.
* Flushing tables Поток запускает команду FLUSH TABLES и ожидает, пока все потоки закроют свои таблицы.
* Killed Кто-то направил команду на закрытие потока, и поток будет закрыт при следующей проверке флага закрытия. Флаг проверяется при каждом основном цикле в MySQL, но в некоторых случаях закрытие потока может занять некоторое время. Если поток заблокирован другим потоком, закрытие будет произведено сразу после того, как другой поток снимет блокировку.
* Sending data Поток обрабатывает строки для оператора SELECT, а также направляет данные клиенту.
* Sorting for group Поток осуществляет сортировку в соответствии с GROUP BY.
* Sorting for order Поток осуществляет сортировку в соответствии с ORDER BY.
* Opening tables Это просто означает, что поток пытается открыть таблицу. Такая процедура осуществляется довольно быстро, если что-либо не мешает открытию. Например, команды ALTER TABLE или LOCK TABLE могут помешать открытию таблицы, пока выполнение команды не будет завершено.
* Removing duplicates Запрос использовал команду SELECT DISTINCT таким образом, что MySQL не смог произвести оптимизацию на начальном этапе. Поэтому MySQL перед отправкой результатов клиенту должен выполнить дополнительное удаление всех дублирующихся строк.
* Reopen table Поток заблокировал таблицу, но обнаружил, что после блокировки структура таблицы изменилась. Он снял блокировку, закрыл таблицу и теперь пытается повторно ее открыть.
* Repair by sorting Код восстановления использует сортировку для создания индексов.
* Repair with keycache Код восстановления использует создание ключей один за другим, через кэш ключей. Это намного медленнее, чем Repair by sorting.
* Searching rows for update Поток осуществляет первую фазу - производит поиск всех совпадающих строк, чтобы затем обновить их. Это действие необходимо выполнить, если команда UPDATE изменяет индекс, который используется для поиска указанных строк.
* Sleeping Поток ожидает, когда клиент направит ему новую команду.
* System lock Поток ожидает получения внешней системной блокировки таблицы. Если не используется несколько серверов mysqld, которые получают доступ к одним и тем же таблицам, системную блокировку можно отключить при помощи параметра --skip-locking.
* Upgrading lock Обработчик INSERT DELAYED пытается заблокировать таблицу, чтобы вставить строки.
* Updating Поток производит поиск строк, которые необходимо обновить, и обновляет их.
* User Lock Поток ожидает GET\_LOCK().
* Waiting for tables Поток получил уведомление, что структура таблицы изменилась, и ему необходимо повторно открыть таблицу, чтобы получить новую структуру. Чтобы повторно открыть таблицу, он должен подождать, пока ее не закроют все остальные потоки. Это уведомление выдается, если другой поток воспользовался командой FLUSH TABLES или к таблице была применена одна из следующих команд: FLUSH TABLES table\_name, ALTER TABLE, RENAME TABLE, REPAIR TABLE, ANALYZE TABLE или OPTIMIZE TABLE. Обработчик INSERT DELAYED завершил работу со всеми вставками и ожидает новые.

Большинство состояний - это очень быстрые операции. Если потоки остаются в любом из этих состояний на протяжении большого количества секунд, то, возможно, возникла какая-то проблема, которую следует устранить.

Существует еще несколько состояний, не упомянутых выше, но большинство из них полезны только для поиска ошибок в mysqld.

**Завдання 4. Отримання інформації про журнали**

SHOW LOGS отображает информацию по состоянию существующих файлов журналов.

* File отображает полный путь к файлу журнала
* Type отображает тип файла журнала (BDB для файлов журналов Berkeley DB)
* Status отображает состояние файла журнала (FREE - если файл может быть удален, или IN USE, если файл необходим подсистеме транзакций).

**Завдання 4. Отримання інформації про надані права доступу**

## SHOW GRANTS

SHOW GRANTS FOR user выводит список команд назначения привилегий, которые нужны при дублировании привилегий для пользователя.

mysql> SHOW GRANTS FOR root@localhost;

+---------------------------------------------------------------------+

| Grants for root@localhost |

+---------------------------------------------------------------------+

| GRANT ALL PRIVILEGES ON \*.\* TO 'root''localhost' WITH GRANT OPTION |

+---------------------------------------------------------------------+

**Завдання 5. Работа с системними таблицями**

Правильно смотреть информацию о базе данных с помощью ее встроенных инструментов. Для в MySQL есть специальная таблица — Information\_schema. Из нее мы и будем брать данные. Но сначала подключитесь к базе данных:

mysql -u root -p

SELECT table\_schema AS "Имя базы данных",  
ROUND(SUM(data\_length + index\_length) / 1024 / 1024, 2) AS "Размер в Мб"  
FROM information\_schema.TABLES  
GROUP BY table\_schema;

Запрос выведет точный размер базы MySQL для каждой из существующих баз, теперь вы можете ориентироваться что и сколько занимает. Если у вас очень большой проект и большая база данных, то вы можете выводить информацию в гигабайтах:

SELECT table\_schema AS "Имя базы данных",  
ROUND(SUM(data\_length + index\_length) / 1024 / 1024 / 1024, 2) AS "Размер в Гб"  
FROM information\_schema.TABLES  
GROUP BY table\_schema;

Но это еще не все. Возможно, в базе данных есть ненужная информация, которая только замедляет работу системы. Мы можем посмотреть размер таблиц для каждой из баз данных. Запрос для этого будет выглядеть следующим образом:

SELECT table\_name AS "Имя таблицы",  
ROUND(((data\_length + index\_length) / 1024 / 1024), 2) AS "Size in (MB)"  
FROM information\_schema.TABLES  
WHERE table\_schema = "wpfc\_options"  
ORDER BY (data\_length + index\_length) DESC;

Данная информация помогает понять какая таблица сколько занимает и это уже может навести на интересные мысли.

## Додатково

<http://code.openark.org/blog/mysql/useful-database-analysis-queries-with-information_schema>

<http://spec-zone.ru/RU/mysql/5.7/information-schema.html>

## Вимоги до звітності:

1. Перелік всіх виконаних запитів
2. Скріншоти результатів із описом найбільш важливих даних про стан сервера, БД, прав доступа та інших параметрів роботи БД.