

FreeBSD

- настройка
- подсчёт трафика
- программы
- почтовые системы
- Шелезяки
- Мелочи
- Файловая система

www

- hosting
- mod_gzip
- mod_log_sql
- mod_geoip
- mod_bandwidth
- mod_accounting
- mod_bunzip2
- mod_mp3
- mod_shapvh
- mod_limitipconn
- mod_sqlinclude
- mod_auth_imap
- webalizer
- Apache + SSL
- auth_ldap
- Настройка AWStats
- apache 2.0
- mod_auth_external
- CMS - TYPO3

[www.lissyara.su](#) -> [статьи](#) -> [FreeBSD](#) -> [www](#) -> [MySQL + кодировки](#)

MySQL + кодировки

Автор: [zg](#).

Использование кодировок в MySQL.

Первым делом я сходил на <http://dev.mysql.com/doc/> и скачал оттуда мануал по 5.1.

Тестовая машинка

```
test# uname -a
FreeBSD test.dm 7.0-RELEASE FreeBSD 7.0-RELEASE #1:
Fri May 9 15:40:21 YEKST 2008      zg@test.dm:/usr/obj/usr/src/sys/GATE  i386
test#
```

Устанавливаем MySQL 5.1

```
test# pkg_add -r mysql51-server
Fetching ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/
packages-7.0-release/Latest/mysql51-server.tbz... Done.
Fetching ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/
packages-7.0-release/All/mysql-client-5.1.22.tbz... Done.
Added group "mysql".
Added user "mysql".

*****

Remember to run mysql_upgrade (with the optional --datadir=<dbdir> flag)
the first time you start the MySQL server after an upgrade from an
earlier version.

*****

test# echo mysql_enable="YES" >> /etc/rc.conf
test# cp /usr/local/share/mysql/my-large.cnf /etc/my.cnf
test# /usr/local/etc/rc.d/mysql-server start
Starting mysql.
test# sockstat | grep mysql
mysql  mysqld    1154  13  tcp4   *:3306          *:
mysql  mysqld    1154  14  stream /tmp/mysql.sock
test#
```



2014-07-27,
[lissyara](#)
[gmirror](#)
Удалённое
создание
софтверного
зеркала
средствами
gmirror, на
диске разбитом
с
использованием
gpart.
Использование
меток дисков
для
монтирования
разделов.

2013-08-20,
[zentarim](#)
[Scan+Print](#)
[server FreeBSD](#)
[9](#)
Настройка
сервера печати
и сервера
сканирования
под управлением
операционной
системы FreeBSD
9 для МФУ Canon
PIXMA MP540

2011-11-20,
[BlackCat](#)
[Разъём на WiFi-](#)
[карту](#)
Делаем съёмной
несъёмную
антену на WiFi-
карте путём

phpBB-2/3
mod_ntlm
mod_ntlm2
Nginx+php+fcgi
OTRS на Apache1
OTRS на Apache20
ApacheStats
mod_evasive
Lighttpd
nginx+php-
fpm+mysql
php + mssql
MySQL + кодировки
svn+apache+trac
php5-oci8
Lighttpd + Apache
CMS Drupal 6.9
Apache22+MySQL
Оптимизация хоста
для CMS
блокировка spam
на www
PostgreSQL 8
Apache 22 + PHP5
+ suPHP
lighttpd +
mod_uploadprogress
VirtualBox +
phpVirtualBox
DNS API
Security
system
Games Servers
X11
Programming

Пускай это не самый "правильный" способ установки MySQL-сервера, зато быстрый и рабочий.

Итак, sockstat показала, что сервер работает, а установка говорит о том, что сервер абсолютно девственный. Чем это грозит? Кодировки по умолчанию выставлены англоязычные, а значит, будут проблемы при использовании кириллицы. Но как это распознать? Проверяем:

```
test# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.1.22-rc-log FreeBSD port: mysql-server-5.1.22

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use test;
Database changed
mysql> create table `test` (`field` VARCHAR(60));
Query OK, 0 rows affected (0.01 sec)

mysql> insert into `test` values ('иван'), ('родил'), ('девчѐнку');
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql>
```

Первым делом используем тестовую базу, которая уже есть на сервере, затем создаём в ней таблицу и вставляем в неё три слова на русском, про кодировки мы пока ничего не знаем и знать не хотим))).

Пока всё хорошо и радужно, никаких ошибок нет, пробуем сделать выборку:

```
mysql> select * from `test`;
+-----+
| field |
+-----+
| иван  |
| родил |
| девчѐнку |
+-----+
3 rows in set (0.01 sec)

mysql> select * from `test` where `field` like "иван";
+-----+
| field |
+-----+
| иван  |
+-----+
1 row in set (0.00 sec)
```

Как видно, запросы работают абсолютно корректно, так где же грабли?... Оказывается мы на них уже

установки ВЧ-
разъёма

2011-09-14,
[manifesto](#)
[Настройка](#)
[git+gitosis](#)
Настройка
системы
контроля версия
исходного кода
в связке
git+gitosis+ssh

подписка

ваш@e-mail

Отправить

СТОИМ:

```
mysql> select * from `test` order by `field` DESC;
+-----+
| field |
+-----+
| девчёнку |
| родил |
| иван |
+-----+
3 rows in set (0.01 sec)
```

Запрос на выборку с обратной сортировкой привёл к тому, что записи просто вывелись в обратном порядке, но не по алфавиту... До удара граблей остаются считанные секунды, но пока растянем удовольствие :) Сперва ответим на вопрос - почему поля не сортируются по алфавиту? У MySQL имеется мощный и богатый механизм для работы с интернациональными наборами символов, но.. но откуда MySQL узнает, что наши символы - есть русский алфавит, мы же качали **английскую** версию? Ничего не остаётся, как идти ковырять мануал на предмет кодировок...

После того, как загрузился 16-метровый мануал, можно не полениться и прочитать первые пару-тройку страниц с оглавлением)), а можно просто сделать поиск на предмет **charset** или **character set**. Не суть важно, но через некоторое время можно найти раздел **9.1.2. Character Sets and Collations in MySQL**, в котором написано много и интересно, а, главное, содержательно про то, каким образом можно и нужно работать с кодировками.

Расставляя точки над и, **Character Set** - транслируется как "кодировка", а **Collation** - сравнение. В чём разница? Сравнение - это правила сравнения букв кодировки. Сравнения работают только в рамках кодировки, и нельзя сравнивать данные в латинице по правилам кириллицы. Поясню на примере: мы, как увидим позже, внесли данные в таблицу на латинице, а сортировать нужно на кириллице, для чего можно использовать ключевое слово **collate**:

```
mysql> select * from `test` order by `field` collate cp1251_general_ci DESC;
ERROR 1253 (42000): COLLATION 'cp1251_general_ci' is not valid for CHARACTER SET 'latin1'
mysql>
```

MySQL отказывается это делать... но почему? Потому, что latin1 не поддерживает сравнение в кириллице, а доступные "сравнения" можно увидеть так:

```
mysql> show collation like 'latin1%';
+-----+-----+-----+-----+-----+-----+
| Collation | Charset | Id | Default | Compiled | Sortlen |
+-----+-----+-----+-----+-----+-----+
| latin1_german1_ci | latin1 | 5 | | Yes | 1 |
| latin1_swedish_ci | latin1 | 8 | Yes | Yes | 1 |
| latin1_danish_ci | latin1 | 15 | | Yes | 1 |
| latin1_german2_ci | latin1 | 31 | | Yes | 2 |
| latin1_bin | latin1 | 47 | | Yes | 1 |
| latin1_general_ci | latin1 | 48 | | Yes | 1 |
```

```
| latin1_general_ci | latin1 | 49 | Yes | 1 |
| latin1_spanish_ci | latin1 | 94 | Yes | 1 |
+-----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql>
```

Ни о какой кириллице не может идти и речи... Куда копать?... В создание таблицы!

```
mysql> show create table `test`;
+-----+-----+
| Table | Create Table |
+-----+-----+
| test  | CREATE TABLE `test` (
  `field` varchar(60) DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1 |
+-----+-----+
1 row in set (0.00 sec)
```

Ага! По-умолчанию при создании таблицы была взята кодировка latin1, значит, если мы изменим таблицу и укажем ей, что надо использовать кириллическую кодировку, то всё заработает?... В мануале написан пример про изменение кодировки таблицы, используем его:

```
mysql> alter table `test` charset "cp1251";
Query OK, 3 rows affected (0.02 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

Ок!
Проверяем, что получилось...

```
mysql> select * from `test` order by `field` collate cp1251_general_ci DESC;
ERROR 1253 (42000): COLLATION 'cp1251_general_ci' is not valid for CHARACTER SET 'latin1'
```

Хм.. опять та же ошибка, но откуда ей взяться?!..

```
mysql> show create table `test`;
+-----+-----+
| Table | Create Table |
+-----+-----+
| test  | CREATE TABLE `test` (
  `field` varchar(60) CHARACTER SET latin1 DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=cp1251 |
+-----+-----+
1 row in set (0.00 sec)
```

Ого, структура таблицы резко изменилась, теперь у неё задана одна кодировка, а у поля совсем другая.. :((Порыв ещё мануал, можно изменить и кодировку столбца:

```
mysql> alter table `test` modify `field` varchar(60) charset "cp1251";
Query OK, 3 rows affected, 3 warnings (0.02 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

```
mysql> show create table `test`;
+-----+-----+
| Table | Create Table |
+-----+-----+
| test  | CREATE TABLE `test` (
  `field` varchar(60) DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=cp1251 |
+-----+-----+
1 row in set (0.01 sec)
```

Ну вот!!! Злой кодировки latin1 нет и в помине, можно проверять наш роддом)))

```
mysql> select * from `test` order by `field` collate cp1251_general_ci DESC;
+-----+
| field |
+-----+
| ??????? |
| ?????? |
| ???    |
+-----+
3 rows in set (0.00 sec)
```

И вот тот страшный удар граблями, который так долго оттягивался! Внимательный читатель мог заметить, что когда была сделана попытка принудительно сменить кодировку столбца, содержащего данные в latin1, то на каждую запись, содержащую русские буквы, у MySQL был варнинг! Это был крик о том, что сервер не знает, каким образом можно перевести данные из latin1 в cp1251, ну и лучшего способа, чем заменить символы не latin1 вопросиками, он не нашёл :))). Роддом безвозвратно потерял потому, что теперь вместо кириллицы в базе содержатся вопросики..

Этого можно было избежать

На самом деле, ситуация, когда изначально выставлена неправильная кодировка, встречается сплошь и рядом. Симптомы можно выявить следующим образом:

```
mysql> show variables like "char%";
+-----+-----+
| Variable_name | Value |
+-----+-----+
| character_set_client | latin1 |
| character_set_connection | latin1 |
| character_set_database | latin1 |
| character_set_filesystem | binary |
| character_set_results | latin1 |
| character_set_server | latin1 |
| character_set_system | utf8 |
+-----+-----+
```

```
| character_sets_dir | /usr/local/share/mysql/charsets/ |
+-----+-----+
8 rows in set (0.02 sec)
```

Именно эти переменные отвечают за дефолтные значения кодировок.

character_set_client - кодировка, в которой данные будут поступать от клиента

character_set_connection - кодировка по умолчанию для всего, что в рамках соединения не имеет кодировки

character_set_database - кодировка по умолчанию для баз

character_set_filesystem - кодировка для работы с файловой системой (LOAD DATA INFILE, SELECT ... INTO OUTFILE, и т.д.)

character_set_results - кодировка, в которой будет выбран результат

character_set_server - кодировка, в которой работает сервер

character_set_system - кодировка, в которой задаются идентификаторы MySQL, всегда UTF8

character_sets_dir - папка с кодировками

Наиболее значимые для простых пользователей следующие переменные: `character_set_client`, `character_set_results`, `character_set_connection`. Поскольку именно они отвечают за внесение, извлечение информации и создание таблиц/баз соответственно. Какими они могут быть?

```
mysql> SHOW CHARACTER SET;
```

Charset	Description	Default collation	Maxlen
dec8	DEC West European	dec8_swedish_ci	1
cp850	DOS West European	cp850_general_ci	1
hp8	HP West European	hp8_english_ci	1
koi8r	KOI8-R Relcom Russian	koi8r_general_ci	1
latin1	cp1252 West European	latin1_swedish_ci	1
latin2	ISO 8859-2 Central European	latin2_general_ci	1
swe7	7bit Swedish	swe7_swedish_ci	1
ascii	US ASCII	ascii_general_ci	1
hebrew	ISO 8859-8 Hebrew	hebrew_general_ci	1
koi8u	KOI8-U Ukrainian	koi8u_general_ci	1
greek	ISO 8859-7 Greek	greek_general_ci	1
cp1250	Windows Central European	cp1250_general_ci	1
latin5	ISO 8859-9 Turkish	latin5_turkish_ci	1
armscii8	ARMSCII-8 Armenian	armscii8_general_ci	1
utf8	UTF-8 Unicode	utf8_general_ci	3
cp866	DOS Russian	cp866_general_ci	1
keybcs2	DOS Kamenicky Czech-Slovak	keybcs2_general_ci	1
macce	Mac Central European	macce_general_ci	1
macroman	Mac West European	macroman_general_ci	1
cp852	DOS Central European	cp852_general_ci	1
latin7	ISO 8859-13 Baltic	latin7_general_ci	1
cp1251	Windows Cyrillic	cp1251_general_ci	1
cp1256	Windows Arabic	cp1256_general_ci	1
cp1257	Windows Baltic	cp1257_general_ci	1

binary	Binary pseudo charset	binary	1
geostd8	GEOSTD8 Georgian	geostd8_general_ci	1

26 rows in set (0.00 sec)

Любую из этих кодировок можно пользоваться на свой вкус. Обычно русскоязычные пользователи предпочитают cp1251 или utf8, но по сути, неважно, в какой кодировке хранятся данные, важно, чтобы она была изначально правильно указана и данные были корректно внесены.

Мануал предлагает нам три варианта задания кодировок:

1. Через names
2. Через непосредственно переменные character_set_*
3. Через настройки самого сервера

ВНИМАНИЕ!!! Первые два варианта работают только в рамках текущего соединения. Это значит, что при следующем подключении все настройки вернутся в начальное состояние! Чтобы не выставлять кодировку каждый раз, нужно воспользоваться третьим вариантом.

Вариант 1 - Через names

```
mysql> set names 'cp1251';
Query OK, 0 rows affected (0.00 sec)

mysql> show variables like 'char%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| character_set_client | cp1251 |
| character_set_connection | cp1251 |
| character_set_database | latin1 |
| character_set_filesystem | binary |
| character_set_results | cp1251 |
| character_set_server | latin1 |
| character_set_system | utf8 |
| character_sets_dir | /usr/local/share/mysqlCharsets/ |
+-----+-----+
8 rows in set (0.02 sec)
```

Ну, тут всё ясно, три самые нужные кодировки в одном)))

Вариант 2 - Через непосредственно переменные character_set_*

```
mysql> set @@character_set_client='cp1251';
Query OK, 0 rows affected (0.00 sec)

mysql> show variables like 'char%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
```

character_set_client	cp1251
character_set_connection	latin1
character_set_database	latin1
character_set_filesystem	binary
character_set_results	latin1
character_set_server	latin1
character_set_system	utf8
character_sets_dir	/usr/local/share/mysql/charsets/

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

8 rows in set (0.01 sec)

Более детальная настройка, чем names.

Вариант 3 - Через настройки самого сервера

Тут можно пойти двумя путями - либо через конфиг файл:

```

---- Файл my.cnf
[client]
# Для местного клиента
default-character-set=cp1251
....

[mysqld]
# Для всего сервера
default-character-set=cp1251
....

```

либо

```
shell> mysqld --character-set-server=cp1251
```

Ещё можно при конфигурировании задать кодировку по умолчанию

```
shell> ./configure --with-charset=latin1
```

Но лучше, когда кодировка настраивается прямо в соединении.

Что делать, если данные внесены в неправильной кодировке

Если база/таблица/данные были созданы/внесены в кодировке отличной от нужной, то необходимо сделать следующее:

1. Создать бэкап базы данных
2. Создать текстовый дамп базы в SQL-запросах (mysqldump или PhpMyAdmin)
3. С помощью текстового редактора исправить вхождения неверной кодировки на нужную (а лучше попросту удалить всю информацию о кодировках и сравнениях)
4. Удалить базу/таблицу
5. Выставить нужную кодировку на клиента/соединение
6. Импортировать данные исправленного SQL-дампа

Этот вариант подходит почти для всех случаев, за исключением некоторых особых ситуаций, например, когда сравнение, выставленное по-умолчанию, не уместно для некоторых полей. Пример - поле для

хранения пароля, необходимо сравнивать его с учётом регистра, тогда как по-умолчанию выставляется сравнение без учёта регистра.

Правильный вариант работы с MySQL

```
mysql> show variables like 'char%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| character_set_client | latin1 |
| character_set_connection | latin1 |
| character_set_database | latin1 |
| character_set_filesystem | binary |
| character_set_results | latin1 |
| character_set_server | latin1 |
| character_set_system | utf8 |
| character_sets_dir | /usr/local/share/mysql/charsets/ |
+-----+-----+
8 rows in set (0.02 sec)
```

Кодировки выставлены неверно, нужно их настроить

```
mysql> set names 'koi8r';
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> show variables like 'char%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| character_set_client | koi8r |
| character_set_connection | koi8r |
| character_set_database | latin1 |
| character_set_filesystem | binary |
| character_set_results | koi8r |
| character_set_server | latin1 |
| character_set_system | utf8 |
| character_sets_dir | /usr/local/share/mysql/charsets/ |
+-----+-----+
8 rows in set (0.02 sec)
```

Я работаю через koi8r, поэтому и выставляю её, но данные в таблице буду хранить в cp1251

```
mysql> create table `test2` (`field` varchar(60)) charset cp1251;
Query OK, 0 rows affected (0.01 sec)
```

Проверяем, всё ли в порядке

```
mysql> show create table `test2`;
+-----+-----+
| Table | Create Table |
+-----+-----+
| test2 | CREATE TABLE `test2` (
```

```

`field` varchar(60) DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=cp1251
+-----+-----+
1 row in set (0.01 sec)

## Вносим данные
mysql> insert into `test2` values ('и раз'), ('два'), ('три'), ('И ять'), ('шесть');
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0

## Проверяем сортировки
## В обычном сравнении "и" и "и" одинаковы, поэтому сравнение идёт до первого отличного символа
mysql> select * from `test2` order by `field` collate cp1251_general_ci ASC;
+-----+
| field |
+-----+
| два  |
| и раз |
| И ять |
| три  |
| шесть |
+-----+
5 rows in set (0.01 sec)

## В бинарном сравнении "и" меньше чем "и", поскольку у неё код меньше
mysql> select * from `test2` order by `field` collate cp1251_bin ASC;
+-----+
| field |
+-----+
| два  |
| И ять |
| и раз |
| три  |
| шесть |
+-----+
5 rows in set (0.00 sec)

```

Таким образом, клиент работает в KOI8-R, но данные хранятся в cp1251, MySQL знает об этом и делает перекодировку на лету.

Ну и на посошок:

```

mysql> set character_set_results='cp1251';
Query OK, 0 rows affected (0.00 sec)

mysql> select * from `test2`;
+-----+
| field |
+-----+
| X пюг |

```

```
| дБЮ |
| РПХ |
| х ЪРЭ |
| ЬЕЯРЭ |
+-----+
```

5 rows in set (0.00 sec)

Выбирать данные можно в любой кодировке, так же, как и вносить, главное - правильно сообщить об этом MySQL.

Итого

Я только пересказал одну главу мануала :)))

размещено: 2008-05-11,
последнее обновление: 2008-06-03,
автор: [zg](#)

Домашний медиацентр

До сих пор смотрите фильмы на компьютере? На телевизоре лучше!



[Комментарии пользователей \[21 шт.\]](#)

[вверх](#)

Статистика сайта

Сейчас на сайте
находится: **7** чел.

За последние 30 мин
было: **45** человек

За сегодня было
525 показов,
144 уникальных IP

Этот информационный блок появился по той простой причине, что многие считают нормальным, брать чужую информацию не уведомляя автора (что не так страшно), и не оставляя линк на оригинал и автора – что более существенно. Я не против распространения информации – только за. Только условие простое – извольте подписывать автора, и оставлять линк на оригинальную страницу в виде прямой, активной, нескриптовой, незакрытой от индексирования, и не запрещенной для следования роботов ссылки.

Если соизволите поставить автора в известность – то вообще почёт вам и уважение.

© lissyara 2006-10-24 08:47 MSK

Время генерации страницы 1.6934 секунд
Из них PHP: 25%; SQL: 75%; Число SQL-запросов: 86 шт.
Исходный размер: 124680; Сжатая: 22042