

Часто, когда разрабатываешь сайт, замечаешь, как на одни и те же грабли наступают разработчики при проектировании базы данных. Сегодня я решил опубликовать свои шпаргалки, на **самые часто встречающиеся ошибки при работе с MySQL**.

Работа с бекапами

Делаем бекап

```
mysqldump -u USER -pPASSWORD DATABASE > /path/to/file/dump.sql
```

Создаём структуру базы без данных

```
mysqldump --no-data -u USER -pPASSWORD DATABASE > /path/to/file/schema.sql
```

Если нужно сделать дамп только одной или нескольких таблиц

```
mysqldump -u USER -pPASSWORD DATABASE TABLE1 TABLE2 TABLE3 > /path/to/file/dump_table.sql
```

Создаём бекап и сразу его архивируем

```
mysqldump -u USER -pPASSWORD DATABASE | gzip > /path/to/outputfile.sql.gz
```

Создание бекапа с указанием его даты

```
mysqldump -u USER -pPASSWORD DATABASE | gzip > `date  
+/path/to/outputfile.sql.%Y%m%d.%H%M%S.gz`
```

Заливаем бекап в базу данных

```
mysql -u USER -pPASSWORD DATABASE < /path/to/dump.sql
```

Заливаем архив бекапа в базу

```
gunzip < /path/to/outputfile.sql.gz | mysql -u USER -pPASSWORD DATABASE
```

или так

```
zcat /path/to/outputfile.sql.gz | mysql -u USER -pPASSWORD DATABASE
```

Создаём новую базу данных

```
mysqladmin -u USER -pPASSWORD create NEWDATABASE
```

Удобно использовать бекап с дополнительными опциями `-Q -c -e`, т.е.

```
mysqldump -Q -c -e -u USER -pPASSWORD DATABASE > /path/to/file/dump.sql, где:
```

-Q оборачивает имена обратными кавычками

-c делает полную вставку, включая имена колонок

-e делает расширенную вставку. Итоговый файл получается меньше и делается он чуть быстрее

Для просмотра списка баз данных можно использовать команду:

```
mysqlshow -u USER -pPASSWORD
```

А так же можно посмотреть список таблиц базы:

```
mysqlshow -u USER -pPASSWORD DATABASE
```

Для таблиц InnoDB надо добавлять **--single-transaction**, это гарантирует целостность данных бекапа.

Для таблиц MyISAM это не актуально, ибо они не поддерживают транзакционность.

[Подробнее](#)

Общие факты

Полезно под каждую базу на боевом сервере создавать своего пользователя

Кодировка базы может быть любой, если она UTF8

В большинстве случаев лучше использовать движок InnoDB

В php лучше забыть про сильно устаревшее расширение mysql и по-возможности использовать pdo или mysqli

Новую копию MySQL всегда можно настроить и оптимизировать

Без особой нужды не стоит открывать MySQL наружу. Вместо этого можно сделать проброс портов
`ssh -fNL LOCAL_PORT:localhost:3306 REMOTE_USER@REMOTE_HOST`

Работа с данными

Числа

На 32-битных системах практически нет смысла ставить для типа INTEGER свойство UNSIGNED, так как такие большие числа в php не поддерживаются.

На 64-битных системах, php поддерживает большие числа, вплоть до MySQL BIGINT со знаком.

Связанные таблицы («Foreign keys») должны иметь полное сходство по структуре ключей. Т.е. если у нас на одной таблице для поля указано «INTEGER UNSIGNED DEFAULT 0 NOT NULL» то и на другой должно быть указано аналогично

Для хранения булевых значений, нужно использовать TINYINT(1)

А деньги лучше хранить в DECIMAL(10, 2), где первое число обозначает количество всех знаков, включая запятую, а второе — количество знаков после запятой. Итого, у нас получится что DECIMAL(10,2) может сохранить 9999999,99

Строки

В старых версиях (до 5.0.3) VARCHAR была ограничена 255 символами, но сейчас можно указывать до 65535 символов

Помните, что тип TEXT ограничен только 64 килобитами, поэтому что бы сохранять «Войну и Мир» пользуйтесь «LONGTEXT»

Самая правильная кодировка для вашей БД UTF8

Даты

Не забывайте, что

DATE, TIME, DATETIME — выводятся в виде строк, поэтому поиск и сравнение дат происходит через преобразование

TIMESTAMP — хранится в виде UNIX_TIMESTAMP, и можно указать автоматически обновлять колонку

Сравнивая типы данных DATETIME и TIMESTAMP, не забывайте делать преобразование типов, например:

```
SELECT * FROM table WHERE `datetime` = DATE(`timestamp`)
```

Перечисления

Для перечислений правильно использовать тип ENUM

Правильно пишется так: ENUM('мама', 'мыла', 'раму')

Можно ставить значение по-умолчанию, как и для любой строки

В базе поле с перечислением хранится как число, поэтому скорость работы — потрясающе высокая

Количество перечислений ~ 65 тысяч

dev.mysql.com/doc/refman/4.1/en/storage-requirements.html
help.scibit.com/mascon/masconMySQL_Field_Types.html

Отладка

Если запросы тормозят, то можно включить лог для медленных запросов в /etc/mysql/my.cnf

А потом оптимизировать запросы через [EXPLAIN](#)

И наблюдать за запросами удобно через программу mytop