



## Урок 7

# Обзор движков MySQL, управление и обслуживание. Подготовка к собеседованию

Что такое движки MySQL и для чего они нужны. Как сделать резервную копию и загрузить схему. Готовимся к поступлению на работу.

### Движки MySQL

MyISAM

Static (Fixed-length)

Dynamic length, формат таблиц

Compressed

Советы

InnoDB

Советы

MERGE

HEAP (MEMORY)

Советы

ARCHIVE

CSV

FEDERATED

BLACKHOLE

Резервная копия и восстановление из бэкапа

[Резервная копия](#)

[Восстановление из резервной копии](#)

[Управление базой данных](#)

[Права пользователей](#)

[Как создать нового пользователя](#)

[Задачи к собеседованию](#)

[Практическое задание](#)

[Дополнительные материалы](#)

[Используемая литература](#)

# Движки MySQL

## MyISAM

- Наличие поддержки транзакций: отсутствует.
- Максимальный размер таблицы на диске: 256 терабайт.
- Тип блокирования: табличный.
- Наличие полнотекстового поиска: есть.
- Поддержка распределения нагрузки при работе в кластере: отсутствует.
- Поддержка целостности данных: отсутствует.
- Поддержка внешних ключей: отсутствует.
- Поддержка реплицирования: есть.
- Лимит на количество индексов: 64.
- Лимит на количество записей в таблице:  $2^{32}$  степени.
- Максимально допустимая длина ключа: 1000 байт.
- Максимальное место под ключи на жестком диске:  $(key\_length+4)/0.67$ .
- Особенности: движок не поддерживает быстрое восстановление, в случае внезапного отказа сами данные сложно восстанавливать.
- При отсутствии разрывов в рядах данных вставки новых записей не блокируют таблицу. Движок поддерживает хранение данных и индексов на разных разделах жесткого диска или на разных устройствах. Каждое поле в таблице может задавать свою кодировку.
- Максимальная сумма длин для полей типов VARCHAR и CHAR: 64 килобайта.

Теперь рассмотрим типы форматов таблиц, основанных на движке MyISAM.

## Static (Fixed-length)

- Применяется автоматически, когда в таблице нет столбцов с типами VARCHAR, VARBINARY, BLOB, TEXT.
- Работает быстрее, стабильнее, оптимизирован для кэширования, потребуется больше места на HDD в сравнении с другими движками.
- При задании типа *принудительно* столбцы типа VARCHAR и CHAR заполняются пробелами, VARBINARY – нулями.

## Dynamic length, формат таблиц

- Все строки с длиной до 4 приводятся к типу VARCHAR.
- Значения пустых строк и значение ноль (0) не будут занимать память (заметьте, что NULL – это не 0!).
- Строка фрагментируется в автоматическом режиме при обновлениях таблицы. Для дефрагментации потребуется регулярная оптимизация таблицы (OPTIMIZE TABLE).
- Таблицы этого типа хуже поддаются восстановлению при сбоях.

## Compressed

- Тип создается при использовании утилиты myisampack.
- Тип только для чтения.

- Тип хорош для крайне медленных средств хранения информации.
- Тип может иметь как фиксированный, так и динамический размер.

## Советы

Опыт разработчиков показывает, что таблицы на движке MyISAM рано или поздно сбоят. К этому надо быть готовым.

- Нельзя принудительно отключать сервер во время работы.
- Нельзя изменять одни и те же данные с разных серверов.
- Нельзя изменять таблицы удаленно и с сервера одновременно.

Так или иначе, MyISAM уже устарел, что признается и самими разработчиками MySQL. Упор нужно делать на более современный движок – InnoDB.

## InnoDB

- Максимальный размер данных на жестком диске: 64 терабайта.
- Наличие поддержки транзакций: полная (четыре уровня изоляции).
- Тип блокирования: строчный + 2 типа блокировок транзакций (SHARED, EXCLUSIVE).
- Наличие полнотекстового поиска: есть, начиная с версии MySQL 5.6.4.
- Поддержка целостности данных: есть.
- Поддержка внешних ключей: есть.
- Особенности: кластеризация индексов для часто выполняемых запросов; есть возможность запуска на базе ОС с лимитированным размером файлов; большой набор настроек для тюнинга производительности; есть возможность применения Raw Disk для таблиц в обход файловой системы; из коробки автоматически завершает транзакции коммитом; автоматическое отслеживание deadlock-ов.

## Советы

В старых версиях (до 5.6) запрос типа `SELECT (*) FROM table` выполняется значительно дольше, чем в MyISAM. Движок не позволяет перенести данные и таблицы копированием файлов БД. Команда `mysqldump` выполняется медленно – лучше использовать `InnoDB Hot Backup`. Движок чувствителен к отсутствию индексов – выигрыш от InnoDB пропадает, если нет индексов.

Начиная с версии 5.6, движок InnoDB является полноценной заменой движку MyISAM.

## MERGE

- Применяется для решения задач слияния нескольких таблиц с одинаковыми структурами в одну.
- Очевидно, таблицы должны иметь идентичную структуру вплоть до порядка столбцов.
- Команда `DROP` не удаляет исходных таблиц.
- Сливаемые таблицы могут располагаться в другой БД.
- Тип применим для создания алиасов (для одной таблицы).
- Не поддерживает полнотекстовый поиск.
- Запрещено смешивать временные и постоянные таблицы.
- Скорость чтения по ключу низкая.
- Отсутствует команда `REPLACE`.
- Структурные изменения исходных таблиц не фиксируются – как следствие, итоговая таблица сломается.

## HEAP (MEMORY)

- Наличие поддержки транзакций: отсутствует.
- Тип блокирования: табличный.
- Поддержка репликации: есть.
- Лимит на размер ключа: 500 байт.
- Данные в таблице теряются при остановке сервера.
- Используемый формат хранения: фиксированная длина данных.
- При удалении данных память не высвобождается.

### Советы

Таблицы больших размеров сбрасываются из оперативной памяти на более медленный жесткий диск. Соответственно, выигрыш в скорости от использования данного движка теряется.

Используется для локальных вычислений, хранения временных данных.

## ARCHIVE

- Максимальный размер таблицы на диске: нет лимитов.
- Тип блокирования: строчный.
- Нет команд DELETE, REPLACE, UPDATE, ORDER BY, а также полей типа BLOB.
- Команда INSERT буферизируется и выполняется с большой задержкой.
- Крайне медленный SELECT.

Применение: журналирование.

## CSV

- Движок сохраняет данные в CSV-формате.
- Есть возможность редактирования таблиц внешними приложениями.
- Слабая документация, много открытых ошибок.

## FEDERATED

Прозрачное соединение с другим сервером БД (не путать с репликацией). Имеет много ограничений. Разработчики планируют создать функционал подключения к серверам БД, отличным от MySQL.

## BLACKHOLE

- Данные сохраняются в никуда.
- Двоичные журналы пишутся.

Применение: оптимизация реплицирования (мастер не сохраняет данные на диск).

# Резервная копия и восстановление из бэкапа

## Резервная копия

Для создания резервной копии вашей БД можно воспользоваться утилитой `mysqldump`.

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

Чтобы сохранить только структуру БД без данных, нужно выполнить команду:

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

Если вы хотите сохранить несколько таблиц:

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

Бэкап может занимать много места. Поэтому его можно сжимать в архив сразу же при создании:

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

Если вы планируете снимать бэкапы регулярно, удобно указывать, когда они были созданы.

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

## Восстановление из резервной копии

Теперь рассмотрим процесс восстановления БД из бэкапа:

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

Если вы используете архив, команда изменится:

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

## Управление базой данных

Для создания новой БД можно воспользоваться утилитой `mysqladmin`:

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

Меняем пароль пользователя-администратора:

```
mysqladmin -u root -p password "newpassword"
```

При помощи ключей к командам можно сделать бэкап более удобным:

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

- -Q – при указании ключа все имена полей будут обрамлены обратными кавычками;
- -c – полная вставка, включая имена колонок;
- -e – расширенная вставка. Файл выходит меньше по размеру и создается чуть быстрее.

Если вы хотите увидеть доступные для пользователя базы, выполните команду:

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

Или же для списка таблиц:

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

Для таблиц на движке InnoDB нужно добавлять ключ `--single-transaction`, который гарантирует целостность бекапа.

Для таблиц MyISAM это уточнение неактуально, так как они не поддерживают транзакции.

## Права пользователей

### Как создать нового пользователя

Давайте начнем с создания нового пользователя из консоли MySQL:

```
CREATE USER 'newuser'@'localhost' IDENTIFIED BY 'password';
```

Получившийся пользователь «newuser» не может делать с базами данных совершенно ничего. Значит, нам нужно предоставить доступ к информации.

```
GRANT ALL PRIVILEGES ON * . * TO 'newuser'@'localhost';
```

Звездочки в этой команде указывают соответственно базу и таблицу, к которым у пользователя будет доступ. Указанная команда даст пользователю права на чтение, редактирование, выполнение любых действий со всеми БД.

После изменения прав доступа всегда обновляйте состояние прав:

```
FLUSH PRIVILEGES;
```

# Задачи к собеседованию

Задача 1. У вас есть социальная сеть, где пользователи (id, имя) могут ставить друг другу лайки. Создайте необходимые таблицы для хранения данной информации. Создайте запрос, который выведет информацию:

- id пользователя;
- имя;
- лайков получено;
- лайков поставлено;
- взаимные лайки.

Задача 2. Для структуры из задачи 1 выведите список всех пользователей, которые поставили лайк пользователям А и В (id задайте произвольно), но при этом не поставили лайк пользователю С.

Задача 3. Добавим сущности «Фотография» и «Комментарии к фотографии». Нужно создать функционал для пользователей, который позволяет ставить лайки не только пользователям, но и фото или комментариям к фото. Учитывайте следующие ограничения:

- пользователь не может дважды лайкнуть одну и ту же сущность;
- пользователь имеет право отозвать лайк;
- необходимо иметь возможность считать число полученных сущностью лайков и выводить список пользователей, поставивших лайки;
- в будущем могут появиться новые виды сущностей, которые можно лайкать.

## Практическое задание

1. Создать нового пользователя и задать ему права доступа на базу данных «Страны и города мира».
2. Сделать резервную копию базы, удалить базу и пересоздать из бекапа.

## Дополнительные материалы

1. «MySQL 5» – Игорь Симдянов, Максим Кузнецов.

## Используемая литература

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. <https://habrahabr.ru/post/64851/>
2. <https://habrahabr.ru/post/105954/>
3. <https://www.digitalocean.com/community/tutorials/mysql-ru>
4. <https://jsehelper.blogspot.com.by/2016/01/sql-1.html>