

# MyISAM устарел: время мигрировать

Sylvain ARBAUDIE · August 4, 2025

MYSQL

MYISAM

INNODB

MIGRATION

## MYISAM IS DEPRECATED — TIME TO MIGRATE

`ALTER TABLE ... ENGINE=InnoDB;` — the best query you'll run this week

### MyISAM — DEPRECATED

- No ACID transactions
- Table-level locking only
- No foreign key constraints
- Frequent corruption on crash
- No active development
- .MYD + .MYI files — legacy format

### InnoDB — DEFAULT ENGINE

- Full ACID transactions
- Row-level locking (MVCC)
- Foreign keys + referential integrity
- Crash recovery via redo log
- Active development + optimization
- FULLTEXT indexes since MariaDB 10.0

`ALTER TABLE `mydb`.`mytable` ENGINE = InnoDB;`

mysql system DB migrated

Temp tables migrated

Zero reasons to stay

MyISAM deprecated — InnoDB is the only path forward

## Конец эпохи

MyISAM официально устарел. Это не сюрприз — это было очевидно годами. Но на этот раз это закреплено в исходном коде MariaDB / MySQL: движок MyISAM помечен как deprecated, а последние бастионы, использовавшие его внутренне, мигрированы.

Системная база данных `mysql` (та, что хранит пользователей, привилегии, таблицы грантов) больше не использует MyISAM. Внутренние временные таблицы тоже. Два последних оправдания для терпимости к MyISAM в продакшне только что исчезли.

## Почему MyISAM выживал так долго

Чтобы понять текущую ситуацию, нужно обратиться к истории MyISAM и его роли в экосистеме.

MyISAM был движком хранения по умолчанию в MySQL до версии 5.5 (2010 год). Более десяти лет он был выбором по умолчанию для миллионов веб-приложений. WordPress, Joomla, Drupal, phpBB — все эти приложения разрабатывались и тестировались преимущественно с MyISAM.

Преимущества MyISAM были реальны в ту эпоху:

- **Простота:** один файл `.MYD` для данных, один файл `.MYI` для индексов. Легко резервировать, легко перемещать.
- **Производительность чтения:** для нагрузок чистого чтения (блоги, сайты-визитки) MyISAM был быстр.
- **Полнотекстовый поиск:** MyISAM поддерживал полнотекстовый поиск задолго до InnoDB.
- **Низкий расход памяти:** MyISAM использовал мало RAM, что было критично во времена серверов с 512 МБ.

Но эти преимущества стали пережитками. InnoDB сегодня предлагает всё это и гораздо больше.

## Почему нужно мигрировать сейчас

---

### Нет транзакций

MyISAM не поддерживает транзакции ACID. Нет `BEGIN`, нет `COMMIT`, нет `ROLLBACK`. Каждый оператор автоматически коммитится. В случае сбоя во время записи ваши данные находятся в неопределённом состоянии.

### Нет блокировки на уровне строки

MyISAM использует блокировку на уровне таблицы. Одна запись блокирует всю таблицу целиком, блокируя все остальные чтения и записи. С InnoDB блокировка происходит на уровне строки, что обеспечивает конкурентность.

### Нет внешних ключей

MyISAM не поддерживает ограничения внешних ключей. Никакой ссылочной целостности на уровне базы. Вы полностью зависите от приложения для поддержания согласованности данных.

### Частая коррумпция

Таблицы MyISAM печально известны своей хрупкостью. Аварийная остановка сервера, полный диск, `kill -9` процесса `mysqld` — и ваши таблицы повреждены. `myisamchk` и `REPAIR TABLE` становятся вашими лучшими друзьями, но они не всемогущи.

## Больше нет активной разработки

Возможно, это самый важный аргумент. Никто больше не работает над MyISAM. Нет исправлений ошибок, нет оптимизаций производительности, нет новых функций. Это замороженный код, накапливающий технический долг.

## Миграция: проще, чем кажется

Хорошая новость в том, что миграция с MyISAM на InnoDB обычно проста.

### Определить таблицы MyISAM

```
SELECT table_schema, table_name, engine, table_rows,  
       ROUND(data_length / 1024 / 1024, 2) AS data_mb  
FROM information_schema.tables  
WHERE engine = 'MyISAM'  
      AND table_schema NOT IN ('mysql', 'information_schema',  
                               'performance_schema', 'sys')  
ORDER BY data_length DESC;
```

### Конвертировать таблицу

```
ALTER TABLE mydb.mytable ENGINE = InnoDB;
```

Вот и всё. MariaDB / MySQL пересоздаёт таблицу с движком InnoDB. Индексы пересоздаются, данные копируются. Для маленьких таблиц это мгновенно. Для больших таблиц это может занять несколько минут.

### На что обратить внимание

Несколько особых случаев для наблюдения при миграции:

**Полнотекстовые таблицы:** если вы используете индексы FULLTEXT на MyISAM, хорошая новость — InnoDB поддерживает индексы FULLTEXT начиная с MySQL 5.6 / MariaDB 10.0. Синтаксис идентичен.

**Таблицы MERGE:** если вы используете движок MERGE (объединение таблиц MyISAM), вам придётся пересмотреть архитектуру. Партиционирование InnoDB или представления — альтернативные решения.

*\*COUNT() без WHERE\*: MyISAM хранит точное количество строк, что делает `SELECT COUNT() FROM table` мгновенным. InnoDB должен сканировать индекс. Если ваше приложение часто делает COUNT(\*)` без условия, вы заметите разницу (минимальную для таблиц менее миллиона строк).*

**Дисковое пространство:** InnoDB использует больше дискового пространства, чем MyISAM для тех же данных (в среднем в 1,5-2 раза больше), в основном из-за MVCC и управления транзакциями. Проверьте доступное пространство перед миграцией.

## Скрипт массовой миграции

Для баз с множеством таблиц MyISAM — систематический подход:

```
-- Сгенерировать команды ALTER TABLE
SELECT CONCAT('ALTER TABLE `', table_schema,`.`', table_name,
              '` ENGINE=InnoDB;') AS migration_sql
FROM information_schema.tables
WHERE engine = 'MyISAM'
      AND table_schema NOT IN ('mysql', 'information_schema',
                              'performance_schema', 'sys')
ORDER BY data_length ASC;
```

Начните с самых маленьких таблиц для валидации процесса, затем переходите к более крупным.

## После миграции

После миграции всех таблиц на InnoDB рекомендуются некоторые настройки конфигурации:

```
# my.cnf
[mysqld]
default_storage_engine = InnoDB
innodb_buffer_pool_size = 70% # от доступной RAM
innodb_log_file_size = 256M   # или больше в зависимости от нагрузки
innodb_flush_log_at_trx_commit = 1 # полная устойчивость
```

И отключите функции MyISAM, которые вам больше не нужны:

```
skip-external-locking
```

```
key_buffer_size = 8M # минимум, для оставшихся системных таблиц
```

## Заключение

---

MyISAM устарел. Это больше не мнение, это технический факт. Системная база мигрировала, временные таблицы мигрировали, код находится в режиме обслуживания без будущего.

Если у вас ещё есть таблицы MyISAM в продакшне, момент мигрировать — сейчас. Конвертация проста, выгоды немедленны (транзакции, блокировка на уровне строки, восстановление после сбоя), а риски оставаться на MyISAM только растут.

```
ALTER TABLE ... ENGINE=InnoDB;
```

 — это лучший запрос, который вы выполните на этой неделе.

---

Эта статья была первоначально опубликована на [Medium](#).