

MyISAM jest przestarzały: czas na migrację

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

Koniec epoki

MyISAM jest oficjalnie przestarzały. To nie niespodzianka — było to oczywiste od lat. Ale tym razem jest to zapisane w kodzie źródłowym MariaDB / MySQL: silnik MyISAM jest oznaczony jako deprecated, a ostatnie bastiony korzystające z niego wewnątrz zostały zmigrowane.

Systemowa baza danych `mysql` (ta przechowująca użytkowników, uprawnienia, tabele grantów) nie używa już MyISAM. Wewnętrzne tabele tymczasowe również nie. Dwie ostatnie wymówki tolerowania MyISAM na produkcji właśnie zniknęły.

Dlaczego MyISAM przetrwał tak długo

Aby zrozumieć obecną sytuację, trzeba cofnąć się do historii MyISAM i jego roli w ekosystemie.

MyISAM był domyślnym silnikiem przechowywania MySQL aż do wersji 5.5 (2010). Przez ponad dekadę był domyślnym wyborem dla milionów aplikacji webowych. WordPress, Joomla, Drupal, phpBB — wszystkie te aplikacje były rozwijane i testowane głównie z MyISAM.

Zalety MyISAM były realne w tamtych czasach:

- **Prostota:** plik `.MYD` dla danych, plik `.MYI` dla indeksów. Łatwy do backupu, łatwy do przeniesienia.

- **Wydajność odczytu:** dla obciążeń czysto odczytowych (blogi, strony wizytówkowe) MyISAM był szybki.
- **Wyszukiwanie pełnotekstowe:** MyISAM obsługiwał wyszukiwanie pełnotekstowe na długo przed InnoDB.
- **Niski ślad pamięciowy:** MyISAM zużywał mało RAM, co było kluczowe w epoce serwerów z 512 MB.

Ale te zalety stały się relikdami. InnoDB oferuje dziś to wszystko i znacznie więcej.

Dlaczego musisz migrować teraz

Brak transakcji

MyISAM nie obsługuje transakcji ACID. Brak `BEGIN`, brak `COMMIT`, brak `ROLLBACK`. Każda instrukcja jest auto-commitowana. W przypadku awarii podczas zapisu dane są w nieokreślonym stanie.

Brak blokowania na poziomie wiersza

MyISAM używa blokowania na poziomie tabeli. Pojedynczy zapis blokuje całą tabelę, blokując wszystkie inne odczyty i zapisy. Z InnoDB blokowanie odbywa się na poziomie wiersza, umożliwiając współbieżność.

Brak kluczy obcych

MyISAM nie obsługuje ograniczeń kluczy obcych. Brak integralności referencyjnej na poziomie bazy. Zależysz całkowicie od aplikacji w utrzymaniu spójności danych.

Częsta korupcja

Tabele MyISAM są notorycznie kruche. Nagłe zatrzymanie serwera, pełny dysk, `kill -9` procesu `mysqld` — i Twoje tabele są uszkodzone. `myisamchk` i `REPAIR TABLE` stają się Twoimi najlepszymi przyjaciółmi, ale nie są niezawodne.

Brak aktywnego rozwoju

To może najważniejszy argument. Nikt już nie pracuje nad MyISAM. Brak poprawek błędów, brak optymalizacji wydajności, brak nowych funkcji. To zamrożony kod gromadzący dług techniczny.

Migracja: prostsza niż się wydaje

Dobra wiadomość: migracja z MyISAM do InnoDB jest generalnie prosta.

Identyfikacja tabel 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;
```

Konwersja tabeli

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

To wszystko. MariaDB / MySQL odbudowuje tabelę z silnikiem InnoDB. Indeksy są odtwarzane, dane kopiowane. Dla małych tabel to natychmiastowe. Dla dużych tabel może zająć kilka minut.

Punkty uwagi

Kilka szczególnych przypadków do monitorowania podczas migracji:

Tabele full-text: Jeśli używasz indeksów FULLTEXT na MyISAM, dobra wiadomość — InnoDB obsługuje indeksy FULLTEXT od MySQL 5.6 / MariaDB 10.0. Składnia jest identyczna.

Tabele MERGE: Jeśli używasz silnika MERGE (unii tabel MyISAM), musisz przemyśleć architekturę. Partycjonowanie InnoDB lub widoki to alternatywy.

COUNT() bez WHERE: MyISAM przechowuje dokładną liczbę wierszy, co czyni `SELECT COUNT() FROM table` natychmiastowym. InnoDB musi skanować indeks. Jeśli Twoja aplikacja często wykonuje `COUNT(*)` bez warunku, zauważysz różnicę (minimalną dla tabel poniżej miliona wierszy).

Przestrzeń dyskowa: InnoDB zużywa więcej przestrzeni dyskowej niż MyISAM dla tych samych danych (średnio 1,5 do 2x więcej), głównie z powodu MVCC i zarządzania transakcjami. Sprawdź dostępną przestrzeń przed migracją.

Skrypt masowej migracji

Dla baz z wieloma tabelami MyISAM, oto systematyczne podejście:

```
-- Generowanie poleceń 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;
```

Zacznij od najmniejszych tabel, by zwalidować proces, potem przejdź do większych.

Po migracji

Po zmigracji wszystkich tabel do InnoDB zalecane jest kilka korekt konfiguracyjnych:

```
# my.cnf
[mysqld]
default_storage_engine = InnoDB
innodb_buffer_pool_size = 70% # dostępnej RAM
innodb_log_file_size = 256M   # lub więcej w zależności od obciążenia
innodb_flush_log_at_trx_commit = 1 # pełna trwałość
```

I wyłącz funkcjonalności MyISAM, których już nie potrzebujesz:

```
skip-external-locking
key_buffer_size = 8M # minimum, dla pozostałych tabel systemowych
```

Podsumowanie

MyISAM jest przestarzały. To już nie opinia, to fakt techniczny. Baza systemowa została zmigrowana, tabele tymczasowe zostały zmigrowane, kod jest w trybie konserwacji bez przyszłości.

Jeśli nadal masz tabele MyISAM na produkcji, moment na migrację jest teraz. Konwersja jest prosta, korzyści natychmiastowe (transakcje, blokowanie na poziomie wiersza, crash recovery), a

ryzyka pozostawania na MyISAM tylko rosną.

`ALTER TABLE ... ENGINE=InnoDB;` — to najlepsze zapytanie, jakie wykonasz w tym tygodniu.

Ten artykuł został pierwotnie opublikowany na [Medium](#).