

MySQL 8.0 Replikacja: pogodzenie Source/Replica z Master/Slave

Aurélien LEQUOY · April 8, 2026

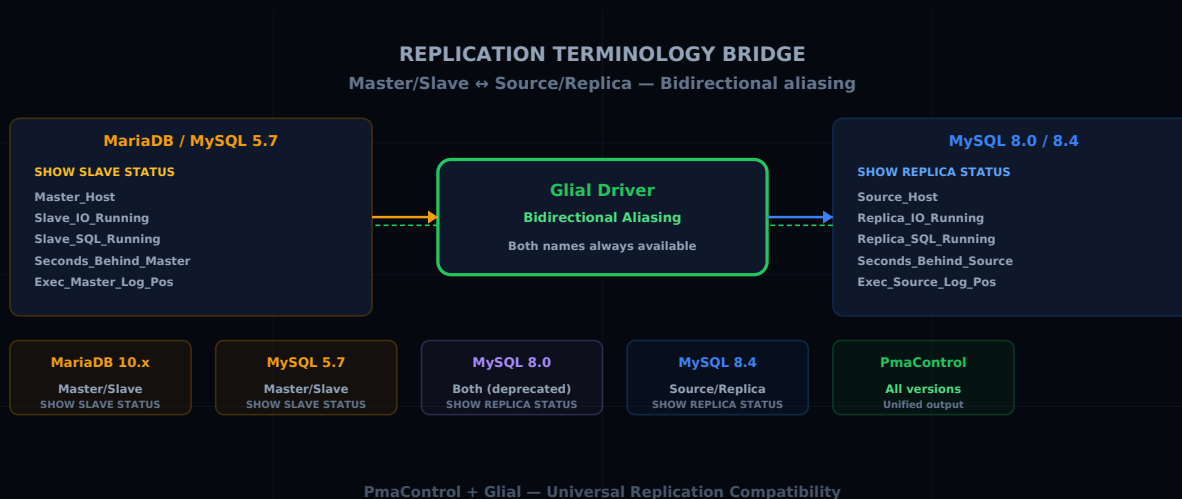
MYSQL

REPLICATION

COMPATIBILITY

MYSQL-8.0

MIGRATION



Przemianowanie

MySQL 8.0.22 wprowadził poważną zmianę terminologiczną: terminy **Master** i **Slave** w poleceniach i kolumnach wyjściowych replikacji zostały zastąpione przez **Source** i **Replica**.

To nie tylko zmiana kosmetyczna. Konkretnie:

- `SHOW SLAVE STATUS` staje się `SHOW REPLICATION STATUS`
- `CHANGE MASTER TO` staje się `CHANGE REPLICATION SOURCE TO`
- Kolumna `Master_Host` w wynikach staje się `Source_Host`
- `Slave_IO_Running` staje się `Replica_IO_Running`
- `Slave_SQL_Running` staje się `Replica_SQL_Running`
- `Seconds_Behind_Master` staje się `Seconds_Behind_Source`

I tak dalej dla około trzydziestu kolumn.

MySQL 8.0 zachowuje kompatybilność wsteczną: stare polecenia nadal działają (z ostrzeżeniem o deprecjacji). Jednak MySQL 8.4 zaczyna usuwać stare aliasy. A MariaDB ze swojej strony

zachowuje historyczną terminologię.

Problem dla narzędzi monitoringu

Każde narzędzie parsujące wyjście `SHOW SLAVE STATUS` / `SHOW REPLICA STATUS` musi teraz obsługiwać **dwa zestawy nazw kolumn** w zależności od wersji serwera:

Stare (MariaDB, MySQL 5.7)	Nowe (MySQL 8.0+)
Master_Host	Source_Host
Master_User	Source_User
Master_Port	Source_Port
Master_Log_File	Source_Log_File
Read_Master_Log_Pos	Read_Source_Log_Pos
Slave_IO_Running	Replica_IO_Running
Slave_SQL_Running	Replica_SQL_Running
Slave_IO_State	Replica_IO_State
Seconds_Behind_Master	Seconds_Behind_Source
Last_IO_Error	Last_IO_Error
Last_SQL_Error	Last_SQL_Error
Exec_Master_Log_Pos	Exec_Source_Log_Pos

Problem jest podwójny:

1. **Istniejący kod** używa starych nazw kolumn wszędzie. Setki odwołań do `$row['Master_Host']` czy `$row['Slave_IO_Running']`.
2. **Infrastruktura jest mieszana**: MariaDB 10.6 i 10.11, MySQL 5.7 u kresu życia, MySQL 8.0, MySQL 8.4 świeżo wdrożony. Wszystko w tym samym PmaControl.

Naiwne rozwiązanie (i dlaczego nie działa)

Pierwszy pomysł: wykrywać wersję i używać właściwej nazwy kolumny.

```
// Nie rób tego
if ($version >= '8.0.22') {
    $host = $row['Source_Host'];
    $io_running = $row['Replica_IO_Running'];
} else {
    $host = $row['Master_Host'];
    $io_running = $row['Slave_IO_Running'];
}
```

Ten wzorzec jest katastrofalny w utrzymaniu. Każde miejsce w kodzie, które odczytuje pole replikacji, musi być zduplikowane. Przy 30 kolumnach i dziesiątkach miejsc w kodzie dochodzi się do setek warunków.

A gdy MySQL 9.0 doda nowe aliasy? Potrojenie kodu?

Podejście Glial: dwukierunkowe aliasowanie na poziomie drivera

Rozwiązanie zaimplementowane w frameworku Glial (który dostarcza warstwę dostępu MySQL dla PmaControl) jest bardziej eleganckie: **dwukierunkowe aliasowanie na poziomie drivera**.

Gdy driver Glial wykonuje `SHOW REPLICATION STATUS` (lub `SHOW SLAVE STATUS`), wzbogaca wynik dodając **obie nazwy** dla każdej kolumny:

```
// W driverze MySQL frameworka Glial
$replication_aliases = [
    'Master_Host'          => 'Source_Host',
    'Master_User'         => 'Source_User',
    'Master_Port'         => 'Source_Port',
    'Master_Log_File'     => 'Source_Log_File',
    'Read_Master_Log_Pos' => 'Read_Source_Log_Pos',
    'Slave_IO_Running'    => 'Replica_IO_Running',
    'Slave_SQL_Running'   => 'Replica_SQL_Running',
    'Slave_IO_State'      => 'Replica_IO_State',
    'Seconds_Behind_Master' => 'Seconds_Behind_Source',
    'Exec_Master_Log_Pos' => 'Exec_Source_Log_Pos',
    // ... wszystkie pary
];
```

```
// Po pobraniu wyniku
foreach ($replication_aliases as $old => $new) {
    if (isset($row[$old]) && !isset($row[$new])) {
        $row[$new] = $row[$old];
    }
    if (isset($row[$new]) && !isset($row[$old])) {
        $row[$old] = $row[$new];
    }
}
}
```

Rezultat: **kod konsumujący może używać dowolnej nazwy — starej lub nowej**. Niezależnie od tego, czy serwer to MariaDB 10.6, MySQL 5.7, MySQL 8.0 czy MySQL 8.4, obie nazwy zawsze istnieją w wyniku.

Zalety tego podejścia

1. Zero modyfikacji istniejącego kodu

Setki odwołań do `$row['Master_Host']` w PmaControl nadal działają. Żadna migracja kodu nie jest potrzebna.

2. Nowy kod może używać nowej terminologii

Programiści piszący nowy kod mogą od razu używać `$row['Source_Host']`. Gdy stary kod będzie stopniowo refaktoryzowany, przejście będzie transparentne.

3. Jeden punkt konserwacji

Aliasowanie jest zdefiniowane raz, w driverze. Jeśli MySQL 9.0 doda nowe pola lub przemianuje inne kolumny, wystarczy rozszerzyć tablicę aliasów.

4. Kompatybilność z SHOW SLAVE STATUS i SHOW REPLICA STATUS

Driver najpierw próbuje `SHOW REPLICA STATUS`. Jeśli polecenie się nie powiedzie (MySQL 5.7, MariaDB), stosuje fallback na `SHOW SLAVE STATUS`. W obu przypadkach wynik zawiera oba zestawy nazw.

```
try {
    $result = $db->query('SHOW REPLICA STATUS');
```

```
} catch (Exception $e) {  
    $result = $db->query('SHOW SLAVE STATUS');  
}  
// W każdym przypadku $result zawiera Master_Host ORAZ Source_Host
```

Przypadek multi-source

MySQL 8.0 obsługuje replikację multi-source (wiele kanałów). Polecenie `SHOW REPLICA STATUS` zwraca wtedy **wiele wierszy**, po jednym na kanał. Aliasowanie jest stosowane do każdego wiersza indywidualnie.

MariaDB również obsługuje replikację multi-source, ale z inną składnią (`SHOW SLAVE 'channel_name' STATUS` lub `SHOW ALL SLAVES STATUS`). Driver Glial normalizuje te różnice, aby PmaControl otrzymywał jednolity format.

Wpływ na PmaControl

Dzięki aliasowaniu w driverze Glial, PmaControl w sposób transparentny obsługuje:

- **MariaDB 10.6 / 10.11:** terminologia Master/Slave, `SHOW SLAVE STATUS`
- **MySQL 5.7:** terminologia Master/Slave, `SHOW SLAVE STATUS`
- **MySQL 8.0:** terminologia Source/Replica, `SHOW REPLICA STATUS` (ze starym fallbackiem)
- **MySQL 8.4:** wyłącznie terminologia Source/Replica, `SHOW REPLICA STATUS`

Interfejs PmaControl wyświetla informacje o replikacji w jednolity sposób, niezależnie od wersji serwera. Dashboard replikacji pokazuje `Source_Host` w kolumnach, ale wewnętrzne zapytania działają z obiema nazwami.

Polecenia również

Przemianowanie dotyczy nie tylko kolumn wyjściowych. Same polecenia SQL również zostały przemianowane:

Stare	Nowe
<code>CHANGE MASTER TO</code>	<code>CHANGE REPLICATION SOURCE TO</code>
<code>START SLAVE</code>	<code>START REPLICA</code>

Stare	Nowe
<code>STOP SLAVE</code>	<code>STOP REPLICA</code>
<code>RESET SLAVE</code>	<code>RESET REPLICA</code>
<code>SHOW SLAVE HOSTS</code>	<code>SHOW REPLICAS</code>

PmaControl stosuje to samo podejście: próba nowego polecenia, fallback na stare.

Rekomendacje dla własnych narzędzi

Jeśli utrzymujesz skrypty lub narzędzia parsujące wyjście replikacji:

1. **Nie używaj warunków opartych na wersji** — to kruche i nie skaluje się
2. **Zaimplementuj dwukierunkowe aliasowanie** na najniższym możliwym poziomie (driver, warstwa abstrakcji)
3. **Pisz nowy kod z nową terminologią** — Source/Replica
4. **Testuj na 4 głównych wersjach**: MariaDB 10.x, MySQL 5.7, MySQL 8.0, MySQL 8.4
5. **Przygotuj się na MySQL 9.x** — stare aliasy mogą całkowicie zniknąć

Podsumowanie

Przemianowanie Master/Slave na Source/Replica w MySQL 8.0 to zmiana prosta z pozoru, ale potencjalnie łamiąca każde narzędzie monitoringu, backupu lub orkiestracji, które parsuje wyjście replikacji.

Właściwym rozwiązaniem jest dwukierunkowe aliasowanie na poziomie drivera: jedna implementacja, zero zmian w kodzie konsumującym, pełna kompatybilność od MariaDB 10.x do MySQL 8.4.

Tak właśnie działa framework Glial i to właśnie pozwala PmaControl nadzorować mieszaną infrastrukturę MariaDB / MySQL bez specjalnej konfiguracji dla poszczególnych wersji.