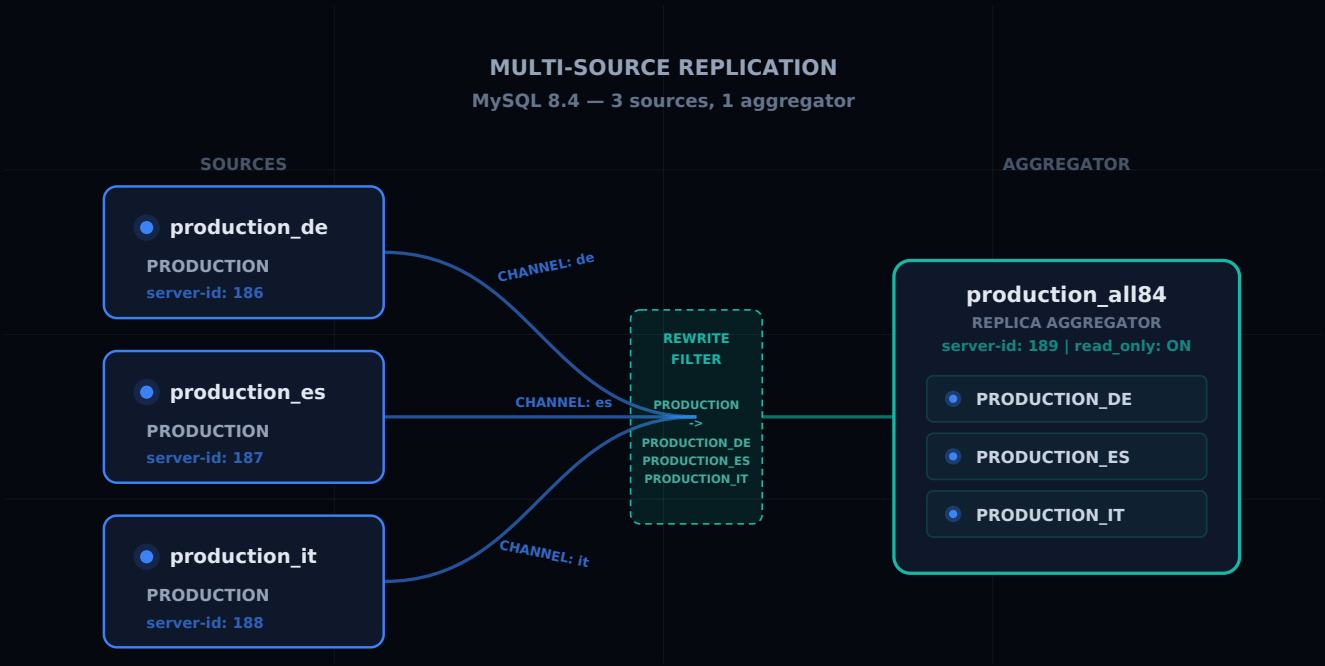
# Replikacja Multi-Source z MySQL 8.4

**Aurélien LEQUOY** · April 12, 2026

MYSQL REPLICATION MULTI-SOURCE MYSQL-8.4




## Wprowadzenie

Replikacja multi-source w MySQL 8.4 pozwala jednej replice otrzymywać transakcje z wielu serwerów źródłowych równolegle. Każde źródło jest podłączone do repliki poprzez odrębny kanał replikacji.

Mechanizm ten służy głównie do:

- konsolidacji wielu serwerów w jednym węźle
- agregacji przepływów z wielu krajów, lokalizacji lub aplikacji
- centralizacji danych do odczytu
- przygotowania węzła raportowego lub rekoncyliacyjnego

Nie jest to jednak klaster zapisu współdzielonego. MySQL nie wykonuje rozwiązywania konfliktów między wieloma źródłami. Jeśli dwa źródła zapisują do tych samych obiektów logicznych, spójność musi być zapewniona po stronie aplikacji lub poprzez ścisłą izolację danych.

## Co MySQL 8.4 faktycznie obsługuje

W MySQL 8.4:

- replika multi-source otwiera jeden kanał na źródło
- każdy kanał musi wskazywać na inne źródło
- replikacja może być oparta na GTID lub na pozycjach binlog
- filtry replikacji mogą być stosowane per kanał
- repozytoria metadanych repliki muszą być w trybie `TABLE`, co jest domyślnym zachowaniem w 8.4

Ważne punkty:

- multi-source służy do konsolidacji, nie do multi-primary z arbitrażem
- nie ma wbudowanego wykrywania ani rozwiązywania konfliktów
- ta sama replika nie może otworzyć wielu kanałów do tego samego źródła

## Przykładowa topologia

Prosty przykład z trzema źródłami i jednym agregatorem:

```
production_de  ─┐
production_es  ─┼──> production_all84
production_it  ─┘
```

W tym przykładzie:

- `production_de` zawiera bazę `PRODUCTION`
- `production_es` również zawiera bazę `PRODUCTION`
- `production_it` również zawiera bazę `PRODUCTION`
- `production_all84` otrzymuje trzy strumienie, ale mapuje je do różnych baz:
  - `PRODUCTION_DE`
  - `PRODUCTION_ES`
  - `PRODUCTION_IT`

To przemapowanie zapobiega zapisywaniu trzech strumieni do tej samej bazy na replice.

## Wymagania wstępne

Na każdym źródle:

- unikatowy `server-id`
- włączony log binarny
- dostęp TCP/IP do portu MySQL
- dedykowany użytkownik replikacji

Na replice multi-source:

- unikatowy `server-id`
- skonfigurowany `relay_log`
- MySQL 8.4
- wstępne przywrócenie danych przed uruchomieniem replikacji

## Typowa konfiguracja źródeł

Przykład minimalnej konfiguracji na źródle:

```
[mysqld]
bind-address = 0.0.0.0
server-id = 186
log_bin = mysql-bin
binlog_format = ROW
binlog_row_image = FULL
sync_binlog = 1
innodb_flush_log_at_trx_commit = 1
```

Ta sama logika na pozostałych źródłach, z innym `server-id` na każdym serwerze.

## Typowa konfiguracja repliki agregatora

Przykład:

```
[mysqld]
bind-address = 0.0.0.0
server-id = 189
```

```
log_bin = mysql-bin
relay_log = mysql-relay-bin
binlog_format = ROW
binlog_row_image = FULL
skip_replica_start = ON
read_only = ON
super_read_only = ON
```

`skip_replica_start=ON` jest przydatny podczas fazy montażu lub utrzymania, ponieważ zapobiega automatycznemu restartowi kanałów przed walidacją.

## Tworzenie konta replikacji

Na każdym źródle:

```
CREATE USER 'repl'@'10.68.68.%' IDENTIFIED BY 'Repl84Geo2026x';
GRANT REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'repl'@'10.68.68.%';
```

Uprawnienie `REPLICATION SLAVE` pozostaje tym udokumentowanym przez MySQL dla tego typu konfiguracji w 8.4.

## Wstępne załadowanie danych

Przed podłączeniem kanału należy załadować dane początkowe na replikę.

Przykład z zrzutami pobranymi ze źródeł:

```
mysqldump --single-transaction --set-gtid-purged=OFF PRODUCTION > PRODUCTION_de.sql
mysqldump --single-transaction --set-gtid-purged=OFF PRODUCTION > PRODUCTION_es.sql
mysqldump --single-transaction --set-gtid-purged=OFF PRODUCTION > PRODUCTION_it.sql
```

Następnie na replice:

```
CREATE DATABASE PRODUCTION_DE;
CREATE DATABASE PRODUCTION_ES;
CREATE DATABASE PRODUCTION_IT;
```

I import trzech zrzutów do trzech odpowiadających im baz docelowych.

## Pobranie współrzędnych binlog

Jeśli nie pracujemy z GTID, należy pobrać dla każdego źródła:

- nazwę pliku binlog
- pozycję startową

Przykład:

```
SHOW BINARY LOG STATUS;
```

Replika rozpocznie następnie od tych współrzędnych z `SOURCE_LOG_FILE` i `SOURCE_LOG_POS`.

## Tworzenie kanałów

W MySQL 8.4 każde źródło konfiguruje się za pomocą `CHANGE REPLICATION SOURCE TO ... FOR CHANNEL`.

Przykład dla źródła niemieckiego:

```
CHANGE REPLICATION SOURCE TO
  SOURCE_HOST='10.68.68.186',
  SOURCE_PORT=3306,
  SOURCE_USER='repl',
  SOURCE_PASSWORD='Repl84Geo2026x',
  SOURCE_LOG_FILE='mysql-bin.000001',
  SOURCE_LOG_POS=158,
  SOURCE_AUTO_POSITION=0,
  GET_SOURCE_PUBLIC_KEY=1
FOR CHANNEL 'production_de';
```

Ta sama zasada dla:

- `production_es`
- `production_it`

## Filtry per kanał

Siła multi-source tkwi w filtrowaniu kanał po kanale.

Jeśli trzy źródła mają bazę `PRODUCTION`, można je przemapować po stronie repliki:

```
CHANGE REPLICATION FILTER
  REPLICATE_REWRITE_DB=((PRODUCTION,PRODUCTION_DE))
FOR CHANNEL 'production_de';

CHANGE REPLICATION FILTER
  REPLICATE_REWRITE_DB=((PRODUCTION,PRODUCTION_ES))
FOR CHANNEL 'production_es';

CHANGE REPLICATION FILTER
  REPLICATE_REWRITE_DB=((PRODUCTION,PRODUCTION_IT))
FOR CHANNEL 'production_it';
```

To jest ważne:

- kanał musi istnieć przed zastosowaniem `CHANGE REPLICATION FILTER ... FOR CHANNEL`
- jeśli kanał jeszcze nie istnieje, MySQL zwraca błąd

## Uruchomienie kanałów

```
START REPLICA FOR CHANNEL 'production_de';
START REPLICA FOR CHANNEL 'production_es';
START REPLICA FOR CHANNEL 'production_it';
```

Następnie, jeśli replika ma pozostać pasywna:

```
SET GLOBAL read_only = ON;
SET GLOBAL super_read_only = ON;
```

## Weryfikacja

Kontrola kanał po kanale:

```
SHOW REPLICA STATUS FOR CHANNEL 'production_de'\G
SHOW REPLICA STATUS FOR CHANNEL 'production_es'\G
SHOW REPLICA STATUS FOR CHANNEL 'production_it'\G
```

Oczekiwane wskaźniki:

- `Replica_IO_Running: Yes`
- `Replica_SQL_Running: Yes`
- `Seconds_Behind_Source: 0` lub bliskie `0`
- `Replicate_Rewrite_DB` poprawnie zdefiniowany

Kontrola funkcjonalna:

```
SELECT * FROM PRODUCTION_DE.germany_feed;
SELECT * FROM PRODUCTION_ES.spain_feed;
SELECT * FROM PRODUCTION_IT.italy_feed;
```

## Typowe operacje

Zatrzymanie pojedynczego kanału:

```
STOP REPLICA FOR CHANNEL 'production_es';
```

Ponowne uruchomienie pojedynczego kanału:

```
START REPLICA FOR CHANNEL 'production_es';
```

Zresetowanie kanału:

```
RESET REPLICA ALL FOR CHANNEL 'production_es';
```

Usunięcie filtra przepisywania na kanale:

```
CHANGE REPLICATION FILTER REPLICATE_REWRITE_DB=() FOR CHANNEL 'production_es';
```

## Czego unikać

- kierowanie wielu źródeł do tej samej bazy docelowej bez izolacji
- zakładanie, że MySQL rozwiąże konflikty kluczy lub kolejności
- używanie różnych wersji MySQL bez ścisłej kontroli kompatybilności
- zapominanie, że `SOURCE_PASSWORD` w `CHANGE REPLICATION SOURCE TO` ma ograniczoną długość
- stosowanie filtrów przed utworzeniem kanału

## Pozycjonowanie techniczne

Multi-source MySQL 8.4 jest odpowiedni do:

- konsolidacji wielu krajów
- scentralizowanego raportowania
- rekoncyliacji
- odbioru wielu niezależnych strumieni

Nie jest odpowiedni, sam w sobie, do:

- prawdziwego multi-master z jednoczesnym zapisem
- architektury konsensusu
- automatycznego rozwiązywania konfliktów

## Podsumowanie

Z MySQL 8.4 replikacja multi-source jest czysta, dojrzała i gotowa do eksploatacji w celu agregacji wielu serwerów źródłowych do jednej repliki.

Zalecany schemat jest prosty:

1. wyraźne rozdzielenie ról źródła i agregatora
2. wymuszenie unikatowego `server-id` wszędzie
3. załadowanie początkowego snapshota
4. utworzenie kanału na źródło
5. zastosowanie filtrów przepisywania per kanał
6. niezależna weryfikacja każdego kanału

Jeśli dane ze źródeł mają identyczne nazwy baz, `REPLICATE_REWRITE_DB` per kanał jest najbardziej przydatnym mechanizmem do utrzymania czytelnej i użytecznej repliki końcowej.

## Oficjalne źródła

- MySQL 8.4 — Multi-Source Replication
- Configuring Multi-Source Replication
- Adding Binary Log Based Sources

- Channel Based Filters
- CHANGE REPLICATION SOURCE TO