Klient mysql i połączenie

Poziom: podstawy **Dotyczy: MySQL 5.7–8.0, MariaDB 10.x–11.x** **Czas: ~20 min**

`mysql` jest tym samym narzędziem od dwudziestu lat — jeden binarny klient wiersza poleceń, cztery parametry połączenia i garść komend zaczynających się od `SHOW`, które robią za brakujący w tym silniku odpowiednik metakomend `psql`.

Skróty użyte na tej stronie (3)

Skrót	Co znaczy
B2B	business to business: sprzedaż i systemy skierowane do innych firm, nie do klientów indywidualnych
CLI	Command Line Interface: praca przez wpisywanie komend w terminalu
TCP	Transmission Control Protocol: protokół sieciowy z potwierdzaniem dostarczenia

Fiszka

TYLKO ODCZYT

Wylistowanie wszystkich baz danych: SHOW DATABASES

```
SHOW DATABASES;
```

Zwracamy listę baz danych widocznych dla zalogowanego użytkownika w całej instancji.

Przykład:

```
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
```

```
| produkcja          |
| sys                |
+--------------------+
5 rows in set (0.00 sec)
```

show = FortiOS: wypisuje konfigurację, w podstawowej postaci tylko ustawienia różniące się od domyślnych, a z wariantem full-configuration także wartości domyślne

TYLKO ODCZYT

Przełączenie się na inną bazę: USE

```
USE NazwaBazy;
```

Ustawiamy bazę domyślną dla bieżącej sesji — kolejne zapytania bez kwalifikacji `NazwaBazy.NazwaTabeli` odnoszą się od teraz do tej bazy.

Przykład:

```
Database changed
```

TYLKO ODCZYT

Wylistowanie tabel bieżącej bazy: SHOW TABLES

```
SHOW TABLES;
```

Zwracamy tabele i widoki z bazy ustawionej wcześniej poleceniem `USE`.

Przykład:

```
+--------------------+
| Tables_in_produkcja |
+--------------------+
| klienci            |
| zamowienia         |
+--------------------+
2 rows in set (0.00 sec)
```

show = FortiOS: wypisuje konfigurację, w podstawowej postaci tylko ustawienia różniące się od domyślnych, a z wariantem full-configuration także wartości domyślne

TYLKO ODCZYT

Wyświetlenie struktury tabeli: DESCRIBE

```
DESCRIBE NazwaTabeli;
```

Zwracamy kolumny tabeli wraz z typem, informacją o `NULL`, kluczach i wartości domyślnej — odpowiednik `\d` w `psql`, tyle że bez indeksów i triggerów, które MySQL pokazuje osobnym poleceniem.

Przykład:

```
+------------+---------+------+-----+---------+----------------+
| Field      | Type    | Null | Key | Default | Extra          |
+------------+---------+------+-----+---------+----------------+
| id         | int(11) | NO   | PRI | NULL    | auto_increment |
| klient_id  | int(11) | NO   | MUL | NULL    |                |
| data       | date    | NO   |     | NULL    |                |
| wartosc    | decimal(10,2) | YES | | NULL  |                |
+------------+---------+------+-----+---------+----------------+
4 rows in set (0.01 sec)
```

ZMIENIA SYSTEM

Wyświetlenie pełnej definicji obiektu: SHOW CREATE TABLE

```
SHOW CREATE TABLE NazwaTabeli;
```

Zwracamy komendę `CREATE TABLE`, którą trzeba by wykonać, żeby odtworzyć tabelę od zera, łącznie z silnikiem, kodowaniem i pełną definicją indeksów oraz kluczy obcych — dokładniejsze niż `DESCRIBE`.

Przykład:

```
*************************** 1. row ***************************
       Table: zamowienia
Create Table: CREATE TABLE `zamowienia` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `klient_id` int(11) NOT NULL,
  `data` date NOT NULL,
```

```
  `wartosc` decimal(10,2) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `idx_klient_id` (`klient_id`),
  CONSTRAINT `zamowienia_ibfk_1` FOREIGN KEY (`klient_id`) REFERENCES `klienci` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
```

□□ **show** = FortiOS: wypisuje konfigurację, w podstawowej postaci tylko ustawienia różniące się od domyślnych, a z wariantem full-configuration także wartości domyślne

□□ TYLKO ODCZYT

Sprawdzenie stanu połączenia i serwera: STATUS

```
STATUS;
```

Zwracamy jednym poleceniem podsumowanie bieżącej sesji — wersję serwera i klienta, sposób połączenia (TCP czy gniazdo), kodowanie, uptime i podstawowe liczniki, bez konieczności odpytywania osobno kilku widoków.

Przykład:

```
--------------
mysql  Ver 15.1 Distrib 10.11.6-MariaDB

Connection id:          482
Current database:       produkcja
Current user:           nazwa_usera@localhost
SSL:                    Not in use
Using delimiter:        ;
Server version:         10.11.6-MariaDB
Protocol version:       10
Connection:              Localhost via UNIX socket
Server characterset:     utf8mb4
Uptime:                  4 days 2 hours 18 min 5 sec

Threads: 12  Questions: 384921  Slow queries: 3  Opens: 210  Flush tables: 1  Open
tables: 64  Queries per second avg: 1.114
--------------
```

□□ **TCP** = Transmission Control Protocol: protokół sieciowy z potwierdzaniem dostarczenia

TYLKO ODCZYT

Wykonanie skryptu SQL z pliku: SOURCE

```
SOURCE SciezkaDoPliku;
```

Wczytujemy i wykonujemy zawartość wskazanego pliku wewnątrz bieżącej sesji, tak jakby jego treść wpisano ręcznie — standardowy sposób odtworzenia zrzutu z `mysqldump`, opisanego na stronie mysqldump i backup logiczny.

Przykład:

```
Query OK, 0 rows affected (0.02 sec)
Query OK, 48213 rows affected (1.84 sec)
Records: 48213  Duplicates: 0  Warnings: 0
```

SQL = Structured Query Language: język zapytań do relacyjnych baz danych · **source** = wykonaj plik w bieżącej powłoce zamiast w nowej, zmienne z niego zostają dostępne dalej

TYLKO ODCZYT

Terminator alternatywny dla wyniku pionowego: \G

```
SELECT * FROM NazwaTabeli WHERE id = 1\G
```

Kończymy polecenie znakiem `\G` zamiast średnika, żeby wynik wypisał się pionowo, jedna kolumna pod drugą zamiast szerokiej tabeli — odpowiednik `\x` w `psql`, tyle że tu decydujemy o tym przy każdym poleceniu z osobna, a nie przełącznikiem trybu na całą sesję.

Przykład:

```
*************************** 1. row ***************************
       id: 1
klient_id: 42
     data: 2026-01-15
  wartosc: 1250.00
1 row in set (0.00 sec)
```

TYLKO ODCZYT

Ustawienie pagera dla długich wyników: pager

```
pager less -SFX
```

Przekierowujemy długie wyniki przez zewnętrzny pager zamiast wypisywać je od razu na cały ekran — flaga `-S` w `less` wyłącza zawijanie wierszy, przydatne przy szerokich tabelach, które inaczej łamią się nieczytelnie. Wyłączamy pager poleceniem `nopager`.

Przykład:

```
PAGER set to 'less -SFX'
```

Z życia wzięte

Z ŻYCIA WZIĘTE

Sklep B2B na my01 przestaje odpowiadać dla jednego z klientów hurtowych i Tomek testuje port z serwera aplikacji poleceniem `mysql -h localhost`. Połączenie przechodzi, ale to nic nie mówi o dostępności portu 3306 z zewnątrz, bo `-h localhost` w MySQL nadal używa gniazda uniksowego. Dopiero `mysql -h 127.0.0.1` pokazuje, że port faktycznie jest zablokowany przez firewalld na my01.

Parametry połączenia

Połączenie z instancją opisują cztery flagi, wszystkie opcjonalne — pominięte przyjmują wartości domyślne (host lokalny przez gniazdo uniksowe, port `3306`, użytkownik systemowy bez hasła):

```
mysql -h NazwaSerwera -P NumerPortu -u NazwaUsera -p
```

Flaga `-p` bez podanej wartości w tym samym słowie wymusza interaktywne pytanie o hasło — `-pHasloUzytkownika` sklejone bez spacji podaje hasło wprost, co jest wygodne w skryptach, ale zostawia hasło widoczne w liście procesów (`ps aux`) dla każdego użytkownika systemu, dopóki proces trwa. Przykład połączenia do zdalnej instancji z wykonaniem jednego polecenia bez wchodzenia w tryb interaktywny:

```
mysql -h my01 -P 3306 -u nazwa_usera -p -e "SHOW DATABASES;"
```

Flaga `-e` (execute) wykonuje podane polecenie i kończy sesję od razu — bez niej `mysql` zostaje w trybie interaktywnym, czekając na kolejne polecenia z klawiatury.

Plik .my.cnf i sekcja [client]

> **Pojęcie: .my.cnf.** Plik w katalogu domowym użytkownika systemowego, czytany automatycznie przez klienta `mysql` oraz przez `mysqldump` i pozostałe narzędzia CLI silnika przy każdym starcie — dzięki niemu nie powtarzamy flag `-h` / `-u` / `-p` w każdym wywołaniu. Sekcja `[client]` obowiązuje wszystkie narzędzia CLI naraz, sekcja `[mysql]` tylko sam klient interaktywny:

```
[client]
host=my01
port=3306
user=nazwa_usera
password=HasloUzytkownika

[mysql]
pager=less -SFX
```

Trzymanie hasła jawnym tekstem w `.my.cnf` wymaga uprawnień `0600` na pliku — z szerszymi uprawnieniami niektóre dystrybucje klienta ostrzegają albo odmawiają odczytu. Bezpieczniejszą alternatywą jest `mysql_config_editor` opisany niżej, który hasła jawnym tekstem w ogóle nie trzyma.

mysql_config_editor — hasło poza plikiem tekstowym

> **Pojęcie: login-path.** Zamiast trzymać hasło jawnym tekstem w `.my.cnf`, `mysql_config_editor` zapisuje zaszyfrowane (nie zahaszowane — odwracalnie zaszyfrowane, więc to ochrona przed przypadkowym podejrzeniem pliku, nie przed kimś z dostępem do konta systemowego) dane logowania do pliku `.mylogin.cnf` pod nazwaną ścieżką logowania:

```
mysql_config_editor set --login-path=NazwaSciezkiLogowania --host=NazwaSerwera --
user=NazwaUsera --password
```

Po ustawieniu łączymy się, podając samą nazwę ścieżki zamiast kompletu flag:

```
mysql --login-path=NazwaSciezkiLogowania
```

Gniazdo uniksowe kontra TCP

> **Pojęcie: gniazdo uniksowe.** Plik specjalny w systemie plików (domyślnie `/var/lib/mysql/mysql.sock` albo `/run/mysqld/mysqld.sock`, zależnie od dystrybucji), przez który proces lokalny łączy się z `mysqld` bez przechodzenia przez stos sieciowy TCP. Podanie `-h localhost` w MySQL i MariaDB, wbrew intuicji, nadal wymusza gniazdo uniksowe — dopiero `-h 127.0.0.1` albo `-h my01` wymusza połączenie przez TCP, nawet jeśli serwer siedzi na tym samym hoście. To zachowanie różni się od PostgreSQL, gdzie sama obecność `-h` (dowolna wartość) przełącza na TCP.

Różnica ma znaczenie praktyczne przy uprawnieniach: konto `'NazwaUsera'@'localhost'` odpowiada wyłącznie na połączenia przez gniazdo (albo `127.0.0.1` w niektórych konfiguracjach uwierzytelniania), a konto `'NazwaUsera'@'%'` odpowiada na połączenia z dowolnego hosta przez TCP — dwa różne konta opisane szerzej na stronie Użytkownicy i uprawnienia.

Częste błędy

Mylenie -h localhost z rzeczywistym TCP. Podanie `-h localhost` w MySQL i MariaDB nadal łączy przez gniazdo uniksowe, nie przez TCP — kto testuje w ten sposób dostępność portu 3306 z zewnątrz, testuje coś innego, niż mu się wydaje. Do wymuszenia TCP lokalnie potrzeba `-h 127.0.0.1`.

Hasło w -pHasloUzytkownika w skrypcie widocznym dla innych userów. Argument wywołania procesu jest widoczny w `ps aux` dla każdego zalogowanego użytkownika, dopóki polecenie trwa — `mysql_config_editor` albo `.my.cnf` z uprawnieniami `0600` są tu bezpieczniejszym rozwiązaniem.

Powiązane: Architektura MySQL i MariaDB: silniki składowania, pliki, procesy · my.cnf: konfiguracja serwera · Użytkownicy i uprawnienia

Źródła: dokumentacja MySQL 8.0, rozdział „mysql — The MySQL Command-Line Client" · `man 1 mysql`

Revision #41
Created 2026-09-25 00:00:30 UTC by Tomasz Konieczny
Updated 2026-10-26 14:08:16 UTC by Tomasz Konieczny