Użytkownicy i uprawnienia

Konta, hosty, role i konta techniczne.

- Użytkownicy i uprawnienia

Użytkownicy i uprawnienia

Poziom: podstawy **Dotyczy: MySQL 5.7–8.0, MariaDB 10.x–11.x** **Czas: ~25 min**

Konto w MySQL to nie sama nazwa użytkownika — to para nazwa plus host, i pomylenie tych dwóch kont jest najczęstszą przyczyną „hasło dobre, a logowanie i tak nie działa".

Skróty użyte na tej stronie (1)

Skrót	Co znaczy
CLI	Command Line Interface: praca przez wpisywanie komend w terminalu

Fiszka

ZMIENIA SYSTEM

Utworzenie kont z tą samą nazwą na dwóch hostach: CREATE USER

```
CREATE USER 'NazwaUsera'@'localhost' IDENTIFIED BY 'TrudneHaslo';
CREATE USER 'NazwaUsera'@'%' IDENTIFIED BY 'TrudneHaslo';
```

Dwie osobne komendy `CREATE USER` tworzą dwa osobne konta — zmiana hasła jednego z nich (`ALTER USER 'NazwaUsera'@'localhost' ...`) nie dotyka drugiego, co jest częstym źródłem nieporozumienia przy rotacji haseł.

ZMIENIA SYSTEM

Nadanie uprawnień na poziomie globalnym: GRANT ... ON .

```
GRANT SELECT ON *.* TO 'NazwaUsera'@'localhost';
```

Poziom globalny (`*.*`) obejmuje wszystkie obecne i przyszłe bazy naraz — uprawnienia administracyjne (`PROCESS`, `RELOAD`, `SUPER`, `REPLICATION CLIENT`) istnieją wyłącznie na tym poziomie, bo dotyczą serwera jako całości, nie pojedynczej bazy.

□□ ZMIENIA SYSTEM

Nadanie uprawnień na poziomie bazy: GRANT ... ON NazwaBazy.*

```
GRANT SELECT ON NazwaBazy.* TO 'NazwaUsera'@'localhost';
```

□□ ZMIENIA SYSTEM

Nadanie uprawnień na poziomie tabeli: GRANT ... ON NazwaBazy.NazwaTabeli

```
GRANT SELECT ON NazwaBazy.NazwaTabeli TO 'NazwaUsera'@'localhost';
```

□□ ZMIENIA SYSTEM

Nadanie uprawnień na poziomie kolumny: GRANT ... (NazwaKolumny)

```
GRANT SELECT (NazwaKolumny) ON NazwaBazy.NazwaTabeli TO 'NazwaUsera'@'localhost';
```

Poziom kolumny jest najrzadziej używany w praktyce — głównie tam, gdzie jedna kolumna tabeli (na przykład dane płacowe) wymaga innego dostępu niż reszta.

□□ ZMIENIA SYSTEM

Utworzenie konta backupu i nadanie podstawowych uprawnień: CREATE USER + GRANT

```
CREATE USER 'NazwaUzytkownika'@'localhost' IDENTIFIED BY 'TrudneHaslo';
GRANT EVENT, LOCK TABLES, SELECT, RELOAD, SHOW VIEW ON *.* TO
'NazwaUzytkownika'@'localhost';
FLUSH PRIVILEGES;
```

Zalogowani do CLI `mysql`/`mariadb` jako root, tworzymy usera pod backup.

□□ **CLI** = Command Line Interface: praca przez wpisywanie komend w terminalu

ZMIENIA SYSTEM

Nadanie pozostałych uprawnień do backupu: GRANT

```
GRANT PROCESS ON *.* TO 'NazwaUzytkownika'@'localhost';
GRANT LOCK TABLES ON *.* TO 'NazwaUzytkownika'@'localhost';
GRANT SHOW VIEW ON *.* TO 'NazwaUzytkownika'@'localhost';
FLUSH PRIVILEGES;
```

Nadajemy uprawnienia potrzebne do tworzenia backupu. `SELECT` daje prawo odczytu danych, `LOCK TABLES` pozwala na blokady spójności przy zrzucie bez `--single-transaction`, `SHOW VIEW` jest wymagane, żeby `mysqldump` w ogóle zobaczył definicję widoków przy zrzucie struktury, `EVENT` i `RELOAD` (to drugie potrzebne przez `FLUSH TABLES WITH READ LOCK` używane wewnętrznie przy niektórych trybach zrzutu) domykają zestaw pozwalający na pełny, spójny backup logiczny bez nadawania kontu praw zapisu.

ZMIENIA SYSTEM

Utworzenie konta monitoringu z minimalnymi uprawnieniami: CREATE USER + GRANT

```
CREATE USER 'NazwaUzytkownikaMonitoring'@'10.20.10.%' IDENTIFIED BY 'TrudneHaslo';
GRANT PROCESS, REPLICATION CLIENT ON *.* TO 'NazwaUzytkownikaMonitoring'@'10.20.10.%';
GRANT SELECT ON performance_schema.* TO 'NazwaUzytkownikaMonitoring'@'10.20.10.%';
```

Konto używane wyłącznie przez system monitoringu (odpytywanie `SHOW STATUS`, `SHOW PROCESSLIST`, widoków `performance_schema`) nie potrzebuje nawet `SELECT` na danych aplikacji. `PROCESS` odsłania pełną listę procesów wszystkich userów, nie tylko własne, `REPLICATION CLIENT` daje dostęp do `SHOW REPLICA STATUS` opisanego na stronie Replikacja, a `SELECT` na `performance_schema` odsłania metryki wydajnościowe bez dostępu do jakiejkolwiek tabeli aplikacji.

TYLKO ODCZYT

Wylistowanie uprawnień konta: SHOW GRANTS

```
SHOW GRANTS FOR 'NazwaUsera'@'localhost';
```

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

◫◫ ZMIENIA SYSTEM

Odebranie uprawnienia: REVOKE

```
REVOKE SELECT ON NazwaBazy.* FROM 'NazwaUsera'@'localhost';
```

◫◫ ZMIENIA SYSTEM

Utworzenie roli i nadanie jej kontu: CREATE ROLE

```
CREATE ROLE NazwaRoli;
GRANT SELECT, INSERT, UPDATE ON NazwaBazy.* TO NazwaRoli;
GRANT NazwaRoli TO 'NazwaUsera'@'localhost';
SET DEFAULT ROLE NazwaRoli TO 'NazwaUsera'@'localhost';
```

Bez `SET DEFAULT ROLE` nadana rola pozostaje nieaktywna po zalogowaniu, dopóki sesja sama nie wykona `SET ROLE NazwaRoli` — łatwe do przeoczenia przy pierwszym wdrożeniu ról, gdy uprawnienia „są nadane", a konto i tak dostaje `Access denied`.

◫◫ **SET** = Switch Embedded Teaming: zespół kart sieciowych zbudowany w przełączniku wirtualnym Hyper-V; łączy dwie karty w jedno łącze bez ustawień po stronie przełącznika fizycznego, więc bez LACP

◫◫ ZMIENIA SYSTEM

Zmiana hasła konta: ALTER USER

```
ALTER USER 'NazwaUsera'@'localhost' IDENTIFIED BY 'NoweHaslo';
```

◫◫ ZMIENIA SYSTEM

Zmiana hasła z ustaleniem wtyczki uwierzytelniania: ALTER USER ... IDENTIFIED WITH

```
ALTER USER 'NazwaUsera'@'localhost' IDENTIFIED WITH mysql_native_password BY
'TrudneHaslo';
```

TYLKO ODCZYT

Sprawdzenie, jakiej wtyczki używa istniejące konto: SELECT z mysql.user

```
SELECT user, host, plugin FROM mysql.user WHERE user = 'NazwaUsera';
```

Z życia wzięte

Z ŻYCIA WZIĘTE

Zespół Oli prosi o konto do bazy `sklep_b2b` na my01, żeby podłączyć nowy panel raportowy z serwera app01. Zakładamy `'raport_b2b'@'10.20.10.41'`, a nie `'raport_b2b'@'%'`, żeby dostęp działał wyłącznie z tego jednego serwera. Miesiąc później ktoś próbuje połączyć się tym samym kontem z laptopa i dostaje `Access denied`, dokładnie zgodnie z zamiarem.

'user'@'%' i 'user'@'localhost' — dwa różne konta

Pojęcie: konto MySQL. W przeciwieństwie do większości silników, gdzie użytkownik jest identyfikowany samą nazwą, MySQL i MariaDB identyfikują konto parą `'NazwaUsera'@'NazwaHosta'` — ta sama nazwa z różnym hostem to dwa całkowicie osobne konta, z osobnymi hasłami i osobnymi uprawnieniami, nawet jeśli w codziennej mowie mówimy o nich tak, jakby to był jeden użytkownik.

`'NazwaUsera'@'localhost'` odpowiada wyłącznie na połączenia przez gniazdo uniksowe (albo `127.0.0.1`, zależnie od konfiguracji uwierzytelniania), opisane na stronie Klient mysql i połączenie. `'NazwaUsera'@'%'` odpowiada na połączenie z dowolnego hosta po sieci — znak `%` jest tu wieloznacznikiem host-maski, działa też jako fragment, na przykład `'NazwaUsera'@'10.20.10.%'` ogranicza dostęp do jednej podsieci. Jeśli istnieją oba konta naraz, MySQL przy próbie logowania wybiera dopasowanie najbardziej precyzyjne, nie pierwsze z brzegu — konto z konkretnym hostem wygrywa z kontem na `%`.

GRANT — poziomy globalny, bazy, tabeli, kolumny

> **Pojęcie: GRANT.** Polecenie nadające kontu uprawnienie do wykonywania określonych operacji na określonym zakresie obiektów. Zakres deklarujemy notacją `poziom.poziom` po słowie `ON` — cztery poziomy szczegółowości, od najszerszego do najwęższego.

Tworzenie usera na potrzeby backupu CLI

Konto pod automatyczny backup opisany na stronie mysqldump i backup logiczny potrzebuje węższego zestawu uprawnień niż konto administracyjne — nie potrzebuje `SUPER` ani prawa modyfikacji danych, potrzebuje prawa je odczytać, zablokować na czas spójnego zrzutu i przeczytać metadane routines/triggerów.

SHOW GRANTS i REVOKE

> **Pojęcie: SHOW GRANTS.** Zwraca zestaw poleceń `GRANT`, które trzeba by wykonać, żeby odtworzyć dokładnie te same uprawnienia konta — wygodniejsze niż przeglądanie surowych tabel `mysql.user` i `mysql.db`.

> **Pojęcie: REVOKE.** Odwrotność `GRANT` — odbiera wcześniej nadane uprawnienie, ze składnią lustrzaną do `GRANT`.

FLUSH PRIVILEGES — kiedy naprawdę jest potrzebny

> **Pojęcie: FLUSH PRIVILEGES.** Przeładowuje tabele uprawnień z bazy `mysql` do pamięci serwera. Polecenia `CREATE USER`, `GRANT`, `REVOKE` i `ALTER USER` od dawna aktualizują pamięć od razu, same w sobie — `FLUSH PRIVILEGES` za nimi jest zbędne i zostaje w wielu starych skryptach wyłącznie z przyzwyczajenia. Jest naprawdę potrzebny wyłącznie po ręcznej edycji tabel `mysql.user`/`mysql.db` bezpośrednim `INSERT`/`UPDATE`/`DELETE` — praktyce dziś odradzanej na rzecz poleceń `GRANT`/`REVOKE` — oraz po przywróceniu tych tabel ze zrzutu backupu.

Role w MySQL 8

> **Pojęcie: rola (MySQL 8 / MariaDB 10.5+).** Nazwany zestaw uprawnień, który można nadać wielu kontom naraz, bez powtarzania tego samego kompletu `GRANT` dla każdego z nich. Zmiana uprawnień roli propaguje się automatycznie na wszystkie konta, którym rolę wcześniej nadano.

Zmiana hasła i wtyczki uwierzytelniania

> **Pojęcie: wtyczka uwierzytelniania.** Mechanizm decydujący o tym, jak serwer weryfikuje hasło podane przy logowaniu — MySQL 8.0 domyślnie używa `caching_sha2_password`, silniejszego kryptograficznie niż starszy `mysql_native_password`, ale niewspieranego przez część starszych sterowników klienckich (starsze wersje PHP-MySQL, niektóre wersje connectorów .NET) sprzed MySQL 8. MariaDB do dziś domyślnie zostaje przy `mysql_native_password` dla nowych kont, ze względu na szerszą kompatybilność wsteczną.

Częste błędy

Zapomnienie o drugim koncie po zmianie hasła. `ALTER USER 'NazwaUsera'@'localhost'` nie zmienia hasła `'NazwaUsera'@'%'` — jeśli aplikacja łączy się przez sieć, a hasło zmieniono tylko na koncie `localhost`, logowanie aplikacji nadal działa na starym haśle, dopóki ktoś nie zauważy dwóch osobnych kont.

Nadawanie GRANT ALL PRIVILEGES kontu aplikacyjnemu z przyzwyczajenia. Konto aplikacji webowej rzadko potrzebuje `DROP`, `ALTER`, `GRANT OPTION` czy uprawnień administracyjnych — szeroki grant nadany „na wszelki wypadek" zamienia błąd w kodzie aplikacji (na przykład nieopatrzne `DELETE FROM tabela` bez `WHERE`) z incydentu ograniczonego w potencjalne uszkodzenie całej bazy.

Powiązane: Klient mysql i połączenie · mysqldump i backup logiczny · Replikacja

Źródła: dokumentacja MySQL 8.0, rozdział „Access Control and Account Management" · dokumentacja MariaDB, rozdział „User Account Management" · procedura autora (backup CLI)