

# System uprawnień użytkowników

## System uprawnień

Jednym z założeń baz danych, jest możliwość obsługiwanie wielu użytkowników. Po pierwszym uruchomieniu serwera, jedynym zarejestrowanym użytkownikiem jest **root**. Root, to tradycyjna nazwa uniksowego konta, które ma pełną kontrolę nad systemem. To konto, powinno być wykorzystywane tylko do celów administracyjnych. Dla każdego użytkownika MySQL, **należy utworzyć konto i nadać mu hasło dostępu**. Co prawda nadawanie haseł nie jest obowiązkowe, jednak ze względów bezpieczeństwa jest nieodzowne.

MySQL posiada **system uprawnień (przywilejów)**, dzięki któremu każdy użytkownik może wykonywać tylko te operacje, na które mu zezwolił administrator. Podczas rejestracji nowego użytkownika, administrator wyszczególnia czynności, które będzie on mógł wykonywać. Obowiązuje przy tym **zasada najmniejszego przywileju** - użytkownik powinien dysponować minimalnym zasobem uprawnień, tylko takich, które są niezbędne do wykonywania powierzonych mu zadań.

## Poziomy uprawnień

W MySQL rozróżniamy następujące poziomy uprawnień:

- **globalny** - przyznane uprawnienia obowiązują we wszystkich istniejących bazach,
- **baza danych** - uprawnienia w zakresie danej bazy danych,
- **tabela** - w zakresie danej tabeli,
- **kolumna** - w zakresie danej kolumny danej tabeli.

## Typy uprawnień

MySQL wykorzystuje następujące typy uprawnień:

- uprawnienia nadawane użytkownikom,
- uprawnienia administratorów,
- uprawnienia specjalne.

Nikomu, z wyjątkiem administratora, nie należy udostępniać **systemowej bazy mysql**, ponieważ są tam przechowywane identyfikatory oraz hasła wszystkich użytkowników. Uprawnienia użytkowników są związane z konkretnymi poleceniami SQL.

**Tabela.1. Uprawnienia użytkowników**

Upewnienie	Poziom	Opis
SELECT	Tabele, kolumny	Pozwala wyszukiwać rekordy w tabelach
INSERT	Tabele, kolumny	Pozwala wstawiać nowe wiersze w tabelach
UPDATE	Tabele, kolumny	Pozwala zmieniać wartości zapisane w tabeli
DELETE	Tabele	Pozwala usuwać wiersze z tabeli
INDEX	Tabele	Pozwala tworzyć i usuwać indeksy w tabelach
ALTER	Tabele	Pozwala zmieniać strukturę istniejących tabel, np. dodawanie kolumn, zmiany nazw kolumn i tabel, zmiany typów danych w kolumnach
CREATE	Bazy danych, tabele	Pozwala tworzyć nowe tabele i bazy danych
DROP	Bazy danych, tabele	Pozwala usuwać bazy lub tabele

**Tabela.2. Uprawnienia administratorów**

Upewnienie	Opis
CREATE TEMPORARY TABLES	Pozwala administratorowi używać słowo kluczowe TEMPORARY w instrukcji CREATE TABLE
FILE	Pozwala wczytywać dane z plików do tabel i odwrotnie
LOCK TABLES	Pozwala jawnie używać instrukcji LOCK TABLES
PROCESS	Pozwala śledzić procesy wykonywane przez serwer i je przerywać
RELOAD	Pozwala powtórnie załadować tabele zawierające informacje na temat praw dostępu oraz na odświeżenie przywilejów, listy nazw łączących się komputerów, dziennika zdarzeń i tabel
REPLICATION CLIENT	Pozwala używać instrukcję SHOW STATUS na nadawcach i odbiorcach replikacji
REPLICATION SLAVE	Pozwala serwerom będącym odbiorcami replikacji łączyć się z serwerem nadawcą.
SHOW DATABASES	Pozwala odczytywać listę wszystkich baz danych przy użyciu instrukcji SHOW DATABASES. Użytkownicy, którzy nie mają tego uprawnienia, mogą zobaczyć tylko bazy, do których przydzielono im dostęp
SHUTDOWN	Pozwala zakończyć pracę serwera MySQL
SUPER	Pozwala zabijać wątki, należące do dowolnego użytkownika

**Tabela.3. Przywileje specjalne**

Upewnienie	Opis
ALL	Nadaje wszystkie uprawnienia opisane w poprzednich tabelach
USAGE	Nie nadaje żadnych uprawnień. Powoduje zarejestrowanie użytkownika i pozwala mu na zalogowanie się, lecz jakiegokolwiek czynności są dla niego niedostępne. Odpowiednie przywileje są w takiej sytuacji nadawane później

**Użytkownik root posiada wszystkie uprawnienia**

# Rejestrowanie użytkowników

Wykonaj ćwiczenia

## Ćwiczenie.1. Nadawanie hasła użytkownikowi Root. Nadaj hasło użytkownikowi root.

1. Za pomocą Notepad++, otwórz plik `xampp\phpMyAdmin\config.inc.php`.
2. Wpisz hasło **admin123** w wierszu kodu PHP - `$cfg['Servers'][$i]['password'] = 'admin123'`, a następnie zapisz zmiany.
3. Nie otwierając żadnej z baz (dla pewności kliknij napis *phpMyAdmin*), wpisz w zakładce SQL polecenie:

```
SET PASSWORD FOR 'root'@'localhost'= PASSWORD('admin123');
```

a następnie kliknij Wykonaj. Fakt ustanowienia hasła powinien pokazać phpMyAdmin.

4. Ponownie za pomocą Notepad++, otwórz plik `xampp\phpMyAdmin\config.inc.php`, wpisz zamiast `config-cookie` oraz usuń wpisane wcześniej hasło **admin123**: Od tej pory uruchomienie panelu administracyjnego będzie wymagało wpisanie użytkownika `root` oraz hasła **admin123**.

## Polecenie *GRANT* - rejestrowanie nowego użytkownika

Polecenie *GRANT* służy do tworzenia nowych użytkowników i nadawania im uprawnień. Posiada następującą składnię:

```
GRANT przywileje [kolumny]
ON obiekt
TO identyfikator_uzytkownika [IDENTIFIED BY 'haslo']
[REQUIRE opcje_ssl]
[WITH [GRANT OPTION | ograniczenia]];
```

Należy w tym miejscu zaznaczyć, że klauzule pisane w nawiasach kwadratowych mają charakter opcjonalny, tzn. mogą być, ale nie muszą. Poszczególne parametry (pisane kursywą) mają następujące znaczenie:

- Parametr *przywileje* - lista uprawnień, oddzielonych przecinkami.
- Parametr *kolumny* - parametr opcjonalny, można podać nazwę pojedynczej kolumny lub listę nazw oddzielonych przecinkami, do których zostaną zastosowane podane uprawnienia.
- Parametr *obiekt* wskazuje bazę lub tabelę, do której zastosowane zostaną podane uprawnienia. Jeżeli chcemy nadać dane uprawnienia we wszystkich bazach, to parametr *obiekt* powinien przyjąć wartość `*.*`. Wtedy dane uprawnienia nadajemy na poziomie globalnym. Jeżeli nie jest używana żadna baza danych, to stosujemy wartość `*`.

Najczęściej, wskazuje się konkretną bazę oraz:

- wszystkie tabele w bazie - `nazwa_bazy.*`,
- dana tabela w bazie - `nazwa_bazy.nazwa_tabeli`,
- pojedyncze kolumny w danej tabeli - `nazwa_bazy.nazwa_tabeli` oraz nadanie odpowiedniej wartości parametrowi *kolumny*.

Jeżeli podczas wykonywania polecenia używana jest jakaś baza danych, to podanie samej nazwy tabeli, zostanie zinterpretowane, jako nadanie uprawnień tabeli o tej nazwie znajdującej się w tej bazie danych.

- Parametr `identyfikator_uzytkownika` powinien wskazywać identyfikator, za pomocą którego, użytkownik loguje się do serwera MySQL.
- Parametr `haslo` - hasło dostępu podawane podczas logowania do serwera.
- Klauzula `REQUIRE` wskazuje, że użytkownik musi się łączyć poprzez protokół SSL (Secure Sockets Layer), a także wskazać opcje SSL.
- Dodanie opcji `WITH GRANT OPTION` spowoduje, że wskazany użytkownik będzie mógł nadawać innym użytkownikom, takie uprawnienia, jakie sam posiada.

Informacje o uprawnieniach, zapisywane są w czterech tabelach systemowej bazy `mysql`. Tabele te mają nazwy: `mysql.user`, `mysql.db`, `mysql.host`, `mysql.tables_priv` oraz `mysql.columns_priv`. Można zmienić dane wprost w tych tabelach, bez stosowania polecenia `GRANT`.

## Polecenie ***REVOKE*** - odbieranie użytkownikowi uprawnień

Polecenie `REVOKE` służy do odbierania użytkownikom określonych uprawnień. Posiada następującą składnię:

```
REVOKE przywileje [kolumny]
ON obiekt
FROM identyfikator_uzytkownika;
```

### Ćwiczenie.2. Rejestrowanie użytkownika mającego status administratora

Po otwarciu phpMyAdmina - zakładka SQL, otwieramy bazę `samochody`, wpisujemy poniższe polecenie SQL i klikamy **Wykonaj**.

```
GRANT ALL
ON *
TO uczen_admin IDENTIFIED BY 'uczen_admin_123';
```

W wyniku tego polecenia, zostanie zarejestrowany użytkownik o nazwie **uczen\_admin**, zostanie zapisane jego hasło **uczen\_admin\_123** oraz zostaną mu nadane wszystkie uprawnienia w zakresie wszystkich istniejących baz danych. Sprawdzamy na zakładce *Uprawnienia* czy użytkownik **uczen\_admin**, został zarejestrowany.

### Ćwiczenie.3. Odbieranie użytkownikowi wszystkich uprawnień

Jak odebrać wszystkie uprawnienia nadane w poprzednim ćwiczeniu? Należy zastosować polecenie SQL:

```
REVOKE ALL
ON *
FROM uczen_admin;
```

W celu sprawdzenia czy nasz użytkownik, nadal istnieje, możemy utworzyć systemową bazę `mysql` - klikając myszką nazwę na liście baz po lewej okna phpMyAdmina, a następnie zastosować polecenie

```
SELECT * FROM user;
```

Należy zwrócić uwagę, że hasła przechowywane są w postaci zaszyfrowanej.

## Ćwiczenie.4. Usuwanie użytkownika z systemu

Należy usunąć naszego testowego użytkownika `uczen_admin`. Jak to wykonać? Kliknąć zakładkę *SQL*, wpisać polecenie

```
DELETE FROM user  
  
WHERE USER='uczen_admin';
```

i kliknąć *Wykonaj*. Po ponownym zastosowaniu

```
SELECT * FROM user;
```

widać, że nasz testowy użytkownik `uczen_admin` został usunięty z tabeli `user`.

Otwórz bazę `samochody` i sprawdź na zakładce *Użytkownicy*, czy nadal istnieje tam `uczen_admin`. Jeżeli tak, to usuń go ręcznie - zaznacz go na liście, a następnie w polu *Usuń zaznaczonych użytkowników* kliknij *Wykonaj*.

## Ćwiczenie.5. Rejestrowanie użytkownika bazy *samochody*

Użytkownik **klent** będzie użytkownikiem tej bazy.

**Zarejestrowanie nowego użytkownika bez uprawnień.** Otwórz bazę `samochody` i zastosuj następujące polecenie SQL:

```
GRANT USAGE  
ON samochody.*  
TO klient IDENTIFIED BY 'klient123';
```

W wyniku tego polecenia, zostanie zarejestrowany użytkownik bez przywilejów, bazy `samochody`, identyfikowany jako `klient`, z hasłem `klient123`. Sprawdzamy na zakładce *Uprawnienia* czy użytkownik `klient`, został zarejestrowany. Nie powinno tam naszego `klienta`, ponieważ nie ma uprawnień. Sprawdzić więc, czy jest on wyszczególniony w tabeli `user` systemowej bazy `mysql`.

## Ćwiczenie.6. Nadanie uprawnień zarejestrowanemu użytkownikowi klient.

Ponownie otwórz bazę `samochody` i zastosuj następujące polecenie SQL:

```
GRANT SELECT, INSERT, UPDATE, DELETE, INDEX, ALTER, CREATE, DROP  
ON samochody.*  
TO klient;
```

Teraz, nie ma już potrzeby podawania hasła użytkownika `klient`. Lista uprawnień znajduje się po słowie kluczowym `GRANT`. Będą one obowiązywać we wszystkich tabelach bazy `samochody`.

Sprawdź na zakładce *Uprawnienia*, uprawnienia użytkownika `klient`. Rysunek 3\_3\_0\_5b. Nadawanie użytkownikowi uprawnień

**Ograniczenie uprawnień użytkownika `klent`.** Administrator, może zawsze ograniczyć uprawnienia użytkowników. Wykonaj to, stosując następujące polecenie:

```
REVOKE ALTER, CREATE, DROP
ON samochody.*
FROM klent;
```

Sprawdź na zakładce *Uprawnienia*, jakie teraz uprawnienia posiada użytkownik `klent`. Lista uprawnień zmniejszyła się o uprawnienia wymienione po słowie kluczowym `REVOKE`.

### **Ćwiczenie.7. Odebranie użytkownikowi `klent` wszystkich uprawnień.**

W tym celu zastosuj następujący kod SQL:

```
REVOKE ALL
ON samochody.*
FROM klent;
```

Sprawdź na zakładce *Uprawnienia*, czy znajduje się tam użytkownik `klent`. Nie posiada on żadnych uprawnień, więc nie powinien znajdować się na tej zakładce.