SYSTEM UPRAWNIEŃ

SYSTEM UPRAWNIEŃ

- Jednym z założeń baz danych, jest możliwość obsługiwania wielu użytkowników. Po pierwszym uruchomienia serwera, jedynym zarejestrowanym użytkownikiem jest **root**.

SYSTEM UPRAWNIEŃ

- **Root** (ang. korzeń), to tradycyjna nazwa uniksowego konta, które ma **pełną kontrolę nad systemem**. To konto, powinno być wykorzystywane tylko do celów administracyjnych.

SYSTEM UPRAWNIEŃ

- 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.

SYSTEM UPRAWNIEŃ

- MySQL posiada **system uprawnień (przywilejów)**, dzięki któremu każdy użytkownik może wykonywać tylko te operacje, na które mu zezwolił administrator.

SYSTEM UPRAWNIEŃ

- 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.

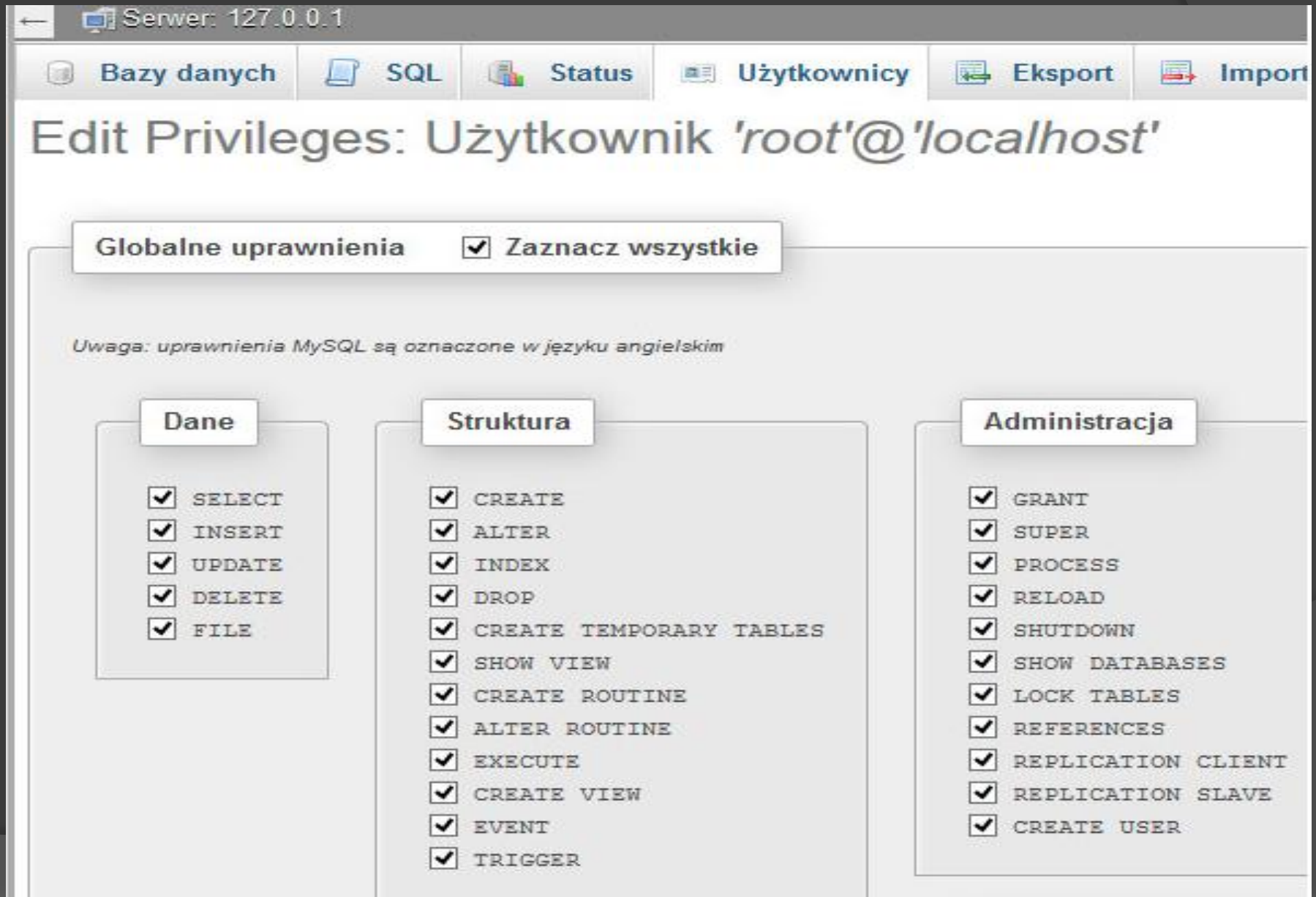
TYPY UPRAWNIEŃ

- MySQL wykorzystuje następujące typy uprawnień:
- uprawnienia nadawane **użytkownikom**,
- uprawnienia **administratorów,**
- uprawnienia **specjalne**.

Systemowa baza mysql

- Nikomu, z wyjątkiem administratora, nie należy udostępniać **systemowej bazy mysql**, ponieważ są tam przechowywane **identyfikatory oraz hasła wszystkich użytkowników**.

Uprawnienia użytkownika root



Nadawanie hasła użytkownikowi *root*

- otwórz plik *xampp\phpMyAdmin\config.inc.php*.
- Wpisz hasło admin123 w wierszu kodu PHP - $cfg['Servers'][$i]['password'] = 'admin123‘
- a wpisz zamiast config - cookie

```
/* Authentication type and info */
$cfg['Servers'][$i]['auth_type'] = 'config';
$cfg['Servers'][$i]['user'] = 'root';
$cfg['Servers'][$i]['password'] = 'admin123';
$cfg['Servers'][$i]['extension'] = 'mysqli';
$cfg['Servers'][$i]['AllowNoPassword'] = true;
$cfg['Lang'] = '';
```

Nadawanie hasła użytkownikowi *root*

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

Uprawnienia użytkowników

Uprawnienie	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

Uprawnienia użytkowników

Uprawnienie	Poziom	Opis
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

Uprawnienia administratorów

Uprawnienie	Opis
CRETE 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 zwierające informacje na temat praw dostępu oraz na odświeżenie przywilejów, listy nazw łączących się komputerów, dziennika zdarzeń i tabel

Uprawnienia administratorów

Uprawnienie	Opis
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

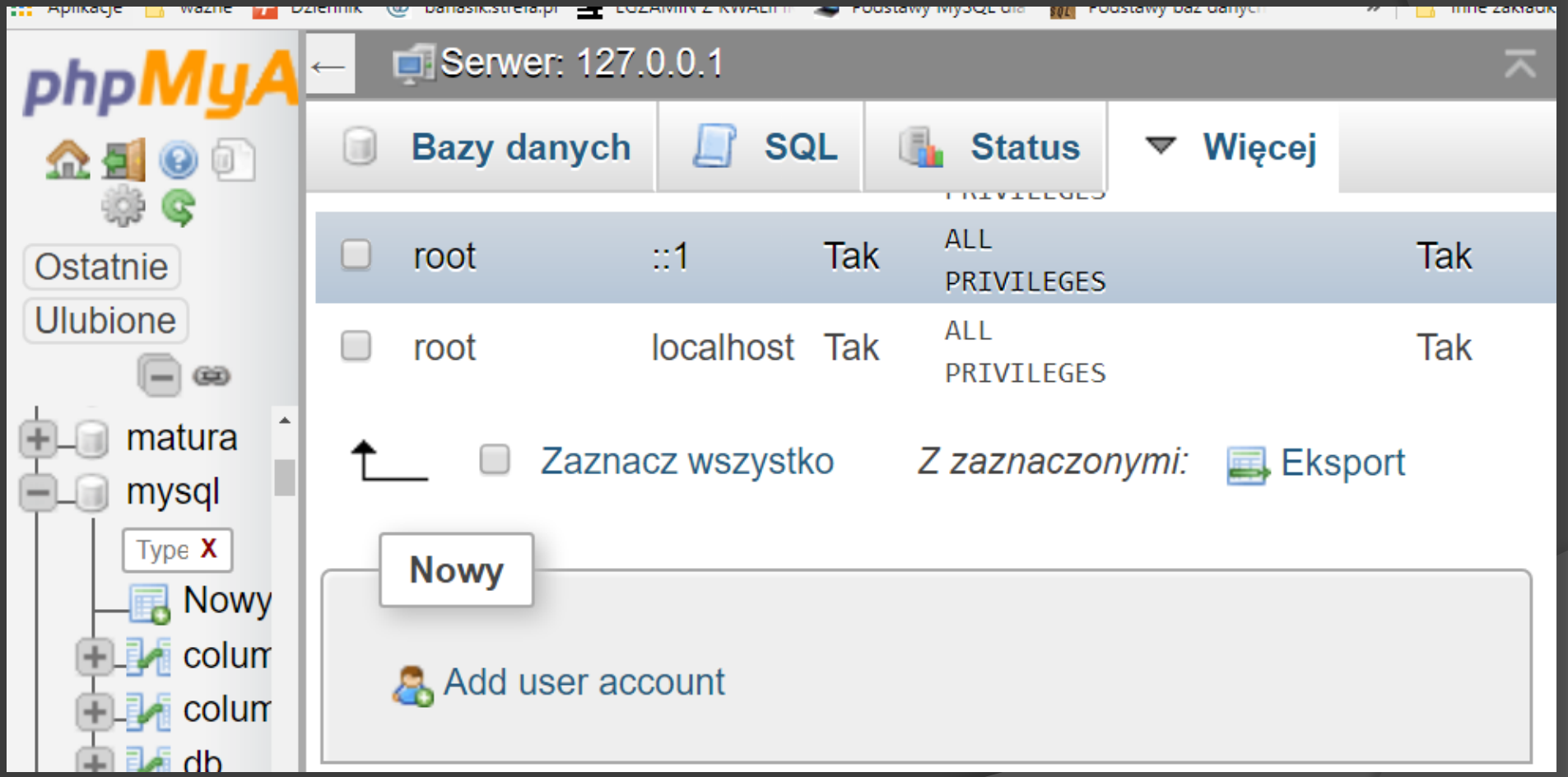
Przywileje specjalne

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

Tworzenie nowego użytkownika

```
CREATE USER uczen_admin@localhost
identified BY 'kasia'
```

Tworzenie nowego użytkownika



Dla dostępu z hosta
o określonej **nazwie**, np. **localhost**:

```
1 CREATE USER 'USER_NAME'@'HOST_NAME'
2 IDENTIFIED BY 'HASŁO';
```

Dla dostępu z hosta o **określonej domenie**, np. *arkadiuszcwiek.pl*:

```
CREATE USER 'USER_NAME'@'HOST_NAME.DOMENA'
IDENTIFIED BY 'HASŁO';
```

Dla dostępu z hosta
o określonym **adresie IP**:

```
CREATE USER 'USER_NAME'@'IP'
IDENTIFIED BY 'HASŁO';
```

Dla dostępu z **dowolnego hosta**:

```
CREATE USER 'USER_NAME'
IDENTIFIED BY 'HASŁO';

co jest równoważne:

CREATE USER 'USER_NAME'@'%'
IDENTIFIED BY 'HASŁO';
```

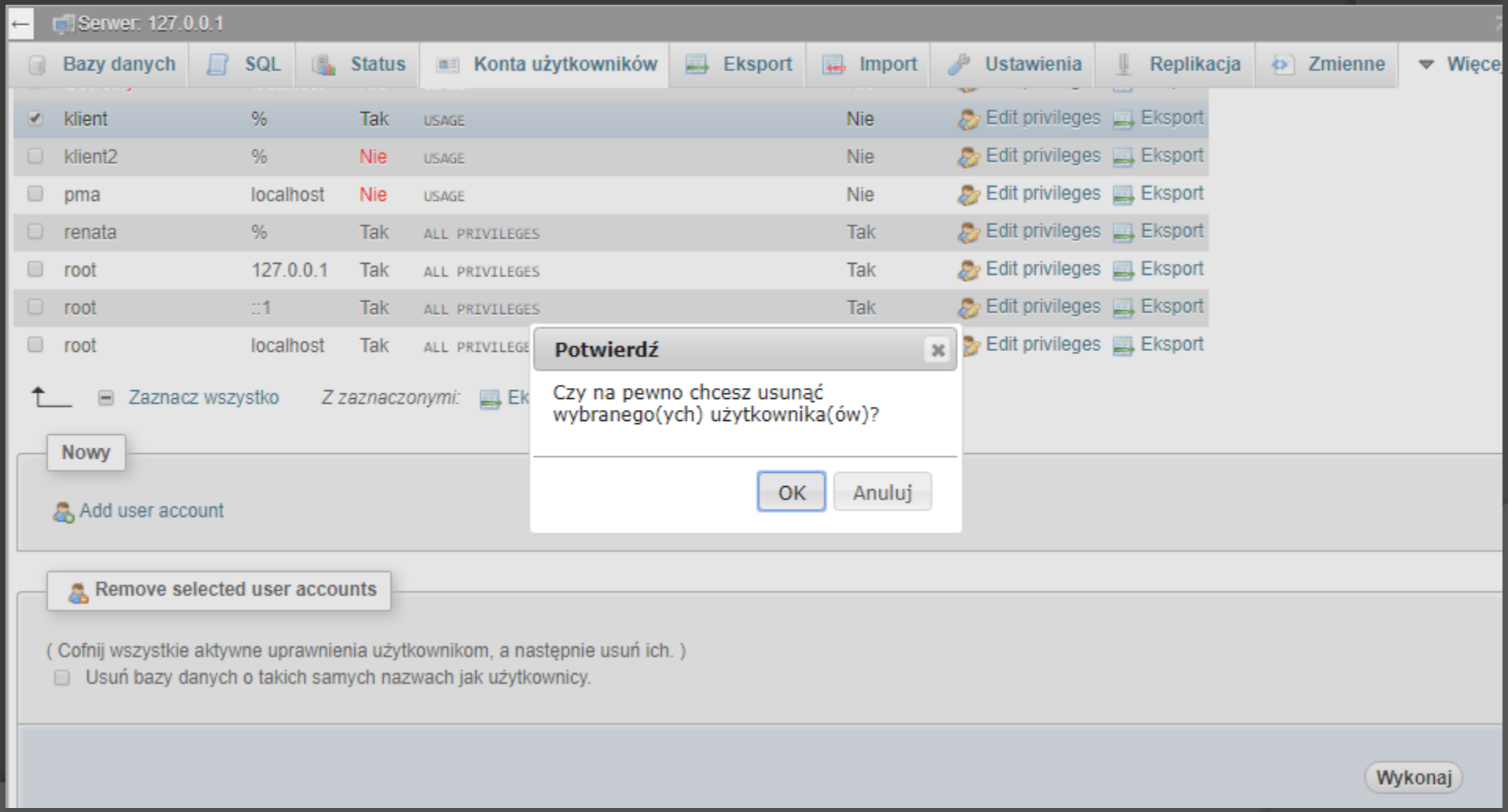
Usuwanie użytkownika

```
1 DROP USER klient
```

Usuwanie użytkownika

```
1 DELETE FROM user
2 WHERE user='uczen_admin'
3
```

Usuwanie użytkownika



Polecenie *GRANT* - nadawanie uprawnień użytkownikowi

- 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**

```
GRANT przywileje [kolumny]
```

- Parametr ***przywileje*** - **lista uprawnień**, oddzielonych przecinkami.

- Parametr ***kolumny*** - parametr opcjonalny, można podać nazwę pojedynczej kolumny lub listę nazw oddzielonych przecinkami,

```
GRANT przywileje
ON obiekt
```

- 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.**,

```
1 GRANT SELECT, INSERT
2 ON studenci.*
3 TO klient
```

- dana tabela w bazie - *nazwa_bazy.nazwa_tabeli,*

```
1 GRANT SELECT, INSERT
2 ON studenci.oceny
3 TO klient
```

- pojedyncze kolumny w danej tabeli - *nazwa_bazy.nazwa_tabeli* oraz nadanie odpowiedniej wartości parametrowi *kolumny.*

```
1 GRANT INSERT(przedmiot)
2 ON studenci.oceny
3 TO klient
```

- 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.

```
TO identyfikator_uzytkownika [IDENTIFIED BY 'haslo']
```

- Parametr *identyfikator_uzytkownika* powinien wskazywać identyfikator, za pomocą którego, użytkownik loguje się do serwera MySQL.
- Parametr *haslo* - hasło dostępu podawane podczas logowania do serwera.

```
[REQUIRE opcje_ssl]
```

- Klauzula REQUIRE wskazuje, że użytkownik musi się łączyć poprzez protokół SSL (Secure Sockets Layer), a także wskazać opcje SSL.

```
[WITH [GRANT OPTION | ograniczenia]];
```

- 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
- mysql.columns_priv.

Można zmienić dane wprost w tych tabelach, bez stosowania polecenia GRANT.

Rejestrowanie nowego użytkownika bez uprawnień

```
GRANT USAGE
ON *
TO klient IDENTIFIED BY 'klient123'
```

Rejestrowanie nowego użytkownika bazy „FIRMA” bez uprawnień

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

Nadanie uprawnień zarejestrowanemu użytkownikowi klient.

```
1 GRANT SELECT, INSERT, UPDATE, DELETE,
2 INDEX, ALTER, CREATE, DROP
3 ON *
4 TO klient
```

Rejestrowanie użytkownika mającego status administratora

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

Przeładowanie nowo nadanych uprawnień

```
FLUSH PRIVILEGES;
```

Przykład.

```
DELETE FROM mysql.user
WHERE User='klient2'
AND Host='%';

FLUSH PRIVILEGES;
```

Przykład 1:

- Tworzenie użytkownika o nazwie „biuro” któremu nadamy prawa tylko do odczytu, wstawiania oraz modyfikowania danych, a więc **SELECT**, **INSERT**, **UPDATE** dla tabeli o nazwie „klienci”.

```
GRANT SELECT, INSERT, UPDATE ON klienci TO biuro
```

Przykład 3:

- W tym przypadku możemy ponownie skorzystać z polecenia **GRANT** dokładając mozliwość usuwania danych dla danego użytkownika.

```
GRANT SELECT, INSERT, UPDATE, DELETE ON klienci TO biuro
```

Przykład 2:

- Tworzymy użytkownika o nazwie „szef” któremu nadajemy wszelkie możliwe uprawnienia dla tabeli „klienci”.

```
GRANT ALL PRIVILEGES ON klienci TO szef
```

Zasada przyznawania minimalnych czyli tylko niezbędnych uprawnień

- Bardzo ryzykowne jest nadanie przywilejów **PROCESS, FILE, SHUTDOWN i RELOAD** użytkownikom którzy nie są administratorami.
- Użytkownik z uprawnieniem **PROCESS**, ma możliwość przeglądania czynności i danych innych użytkowników, w tym również haseł dostępu

Zasada przyznawania minimalnych czyli tylko niezbędnych uprawnień

- natomiast **FILE** uprawnia do edycji plików systemowych serwera.
- Ostrożność jest wskazania przy dodawaniu uprawnienia **GRANT**, ponieważ taki użytkownik może nadawać innym użytkownikom swoje uprawnienia.

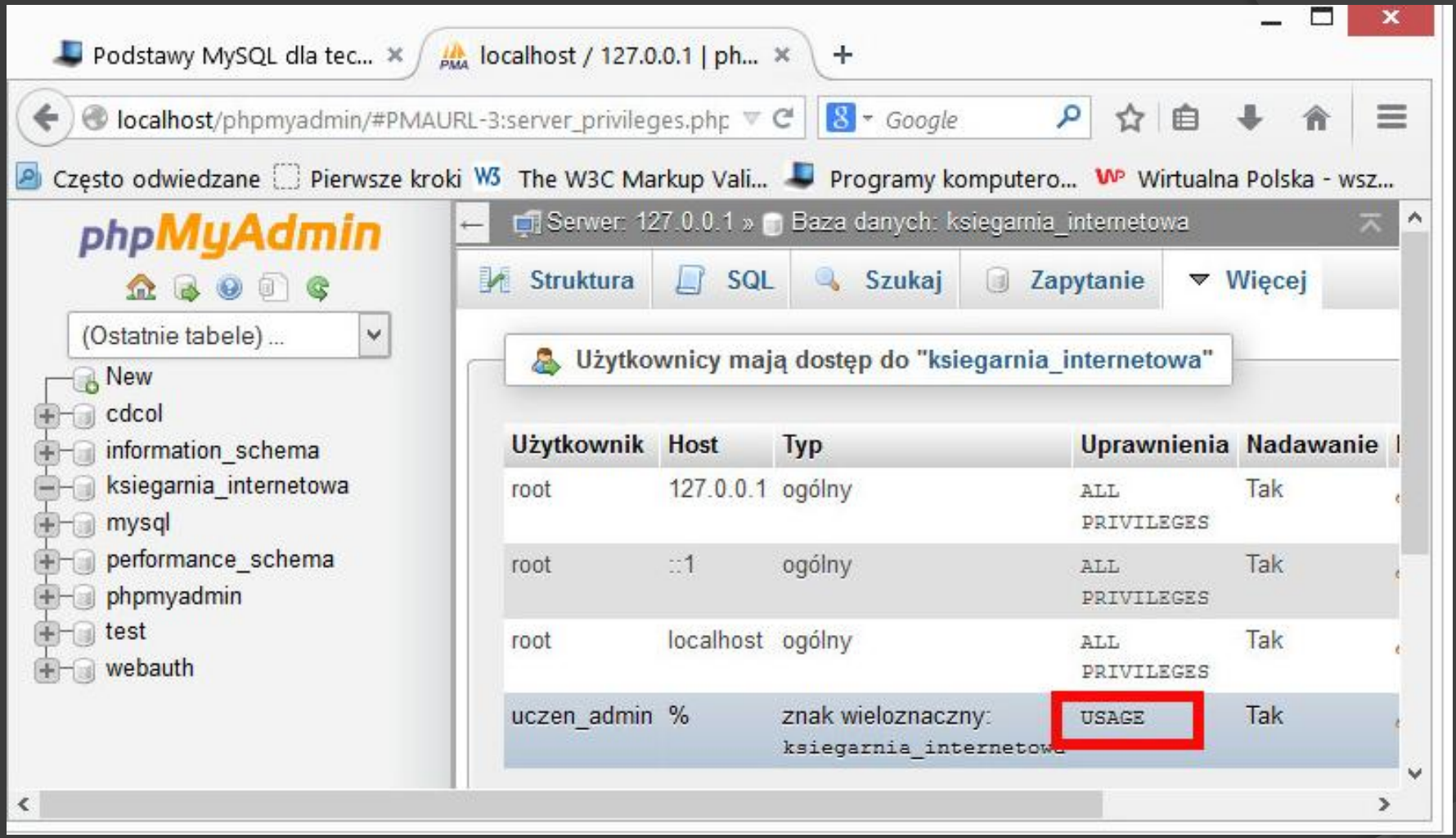
Polecenie *REVOKE* - odbieranie użytkownikowi uprawnień

- służy do odbierania użytkownikom określonych uprawnień. Posiada następującą składnię:

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

Odbieranie użytkownikowi wszystkich uprawnień

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



USAGE zakreślone na rysunku kolorem czerwonym oznacza brak uprawnień danego użytkownika.

Ograniczenie uprawnień użytkownika klient dla bazy firma

```
REVOKE ALTER, CREATE, DROP
ON ksiegarnia_internetowa.*
FROM klient;
```

Ograniczenie uprawnień użytkownika klient dla bazy firma

```
1 REVOKE ALTER, CREATE, DROP
2 ON firma.*
3 FROM klient
```

Zmiana hasła dla root

```
UPDATE user
 SET password=Password('admin')
 WHERE User='root'
```

Zmiana uprawnień globalnych **INSERT** oraz **DELETE** dla użytkownika klient2

```
UPDATE user
SET Insert_priv='Y', Delete_priv='Y'
WHERE User='klient2';
```

Zmiana nazwy użytkownika - instrukcja RENAME USER

```
RENAME USER 'stara_nazwa' TO 'nowa_nazwa';
```

Instrukcja RENAME USER została dodana w MySQL 5.0.2.

Instrukcja SET PASSWORD do nadania lub zmiany hasła użytkownika.

```
SET PASSWORD = PASSWORD('admin''admin123')
```

zmiana hasła dla aktualnie zalogowanego użytkownika.

Zmiany hasła podanego użytkownika

- Aby z niej skorzystać musisz posiadać przywilej UPDATE do bazy mysql.
- Nazwa użytkownika powinna być podana w formacie:

  NazwaUżytkownika@nazwaHosta,

  gdzie obie wartości powinny być identyczne z wyświetlanymi kolumnami User i Host bazy mysql.user.

Zmiany hasła podanego użytkownika

```
SET PASSWORD FOR 'jan'@'%.loc.gov'=PASSWORD('nowe_haso');
```

lub

```
UPDATE mysql.user SET Password=PASSWORD('nowe_haso')
       WHERE User='jan' AND Host='%.loc.gov';
FLUSH PRIVILEGES;
```

Przykłady

```
GRANT SELECT, INSERT ON Osoby TO Tadek

GRANT INSERT(Nazwa) ON Produkty TO Tadek, Ania

GRANT SELECT, UPDATE ON Wyniki TO Piotr WITH GRANT OPTION

GRANT UPDATE(Suma) ON Wyniki TO Iwona

GRANT SELECT, UPDATE ON Wyniki TO Iwona
```

WITH GRANT OPTION - opcją pozwalającą mu nadawać prawa do tej tabeli.

kiedy chcemy zabrać wszystkie możliwe przywileje wymienionym użytkownikom:

- REVOKE ALL PRIVILEGES, GRANT OPTION FROM user1 [, user2] ...

Przykłady

```
REVOKE SELECT, INSERT ON Osoby FROM Tadek;

REVOKE INSERT(Nazwa) ON Produkty FROM Tadek, Ania;

REVOKE GRANT OPTION FOR INSERT ON Produkty FROM Tadek;
```

odebranie praw nadawania uprawnień innym

1. W tabeli artykuly wykonano następujące polecenia dotyczące praw użytkowania jan. Po wykonaniu poleceń użytkownik jan będzie miał prawa do:

```
GRANT ALL PRIVILEGES ON artykuly TO jan
REVOKE SELECT, UPDATE ON artykuly FROM jan
```

A. tworzenia tabeli i aktualizowania w niej danych

B. aktualizowania danych i przeglądania tabeli

C. tworzenia tabeli i wypełniania jej danymi

D. przeglądania tabeli

2. Polecenie nadajace użuykownikowi „szef” wszelkie możliwe uprawnienia dla tabeli „klienci”

A. GRANT PRIVILEGES ON klienci TO szef
B. GRANT ALL PRIVILEGES TO klienci ON szef
C. GRANT ALL PRIVILEGES ON klienci TO szef
D. GRANT ALL PRIVILEGES TO szef

3. Aby usunąć użytkownika ‘kot’ posiadający uprawnienia do przeglądania bazy należy wydać polecenia:

A. DROP kot;
B. SHOW GRANTS kot; DROP kot;
C. GRANT SELECT ON *.* TO kot; DROP kot;
D. REVOKE SELECT ON *.* TO kot; DROP kot;

4. Odświeżenie informacji o użytkownikach bazy danych i ich przywilejach:

A. CLEAR PRIVILEGES;
B. RELOAD PRIVILEGES;
C. REFRESH PRIVILEGES;
D. FLUSH PRIVILEGES;

5. Aby przypisać hasło dla istniejącego użytkownika należy:

A. SET PASSWORD = PASSWORD('jakies haslo');
B. SET PASSWORD ('jakies haslo');
C. SET PASSWORD = ('jakies haslo');
D. PASSWORD('jakies haslo') FOR USER;

6. Aby zabrać wszystkie możliwe przywileje wymienionym użytkownikom należy:

A. REVOKE ALL PRIVILEGES;

B. REVOKE ALL PRIVILEGES, GRANT OPTION FROM user1 [, user2] ...

C. REVOKE ALL PRIVILEGES FROM user1 [, user2] ...

D. DROP ALL PRIVILEGES FROM user1 [, user2] ...

7. Utworzymy użytkownika o nazwie „biuro” któremu nadamy prawa do odczytu, wstawiania oraz modyfikowania danych dla tabeli „klienci”.

A. GRANT SELECT, INSERT, UPDATE ON biuro TO klienci

B. ADD USER biuro GRANT SELECT, INSERT, UPDATE ON klienci

C. GRANT SELECT, INSERT ON biuro TO klienci

D. GRANT SELECT, INSERT, UPDATE ON klienci TO biuro

8. Użytkownik „biuro” nie powinien mieć możliwości wstawiania nowych danych do tabeli „klienci” jak sformułować polecenie odbierające mu takie prawo:

A. REVOKE INSERT ON klienci FROM biuro.

B. DELETE INSERT ON klienci FROM biuro.

C. DELETE FROM INSERT ON klienci FROM biuro.

D. GRANT SELECT, INSERT, UPDATE ON klienci TO biuro

9. Użytkownik „biuro” nie powinien mieć możliwości wstawiania nowych danych do tabeli „klienci” jak sformułować polecenie odbierające mu takie prawo:

A. REVOKE INSERT ON klienci FROM biuro.

B. DELETE INSERT ON klienci FROM biuro.

C. DELETE FROM INSERT ON klienci FROM biuro.

D. GRANT SELECT, INSERT, UPDATE ON klienci TO biuro