Zarządzanie użytkownikami

Poziom: podstawy **Dotyczy: Sybase ASE 15.7 / 16.0 (SAP ASE)** **Czas: ~15 min**

Login na poziomie serwera i user na poziomie bazy danych to dwa osobne obiekty w Sybase — zanim nadamy komukolwiek dostęp, warto rozumieć tę różnicę.

Skróty użyte na tej stronie (2)

Skrót	Co znaczy
ASE	Adaptive Server Enterprise: nazwa silnika Sybase od 1996 roku, dziś rozwijanego przez SAP jako SAP ASE
SAP	Systeme, Anwendungen und Produkte in der Datenverarbeitung: niemiecki producent oprogramowania klasy ERP, właściciel silnika HANA i przejętego wcześniej Sybase ASE

Fiszka

Komendy wykonujemy zalogowani do `isql`, opisanego na stronie isql: klient wiersza poleceń.

TYLKO ODCZYT

Wylistowanie wszystkich userów bieżącej bazy: sp_helpuser

```
sp_helpuser
go
```

Zwracamy listę wszystkich userów zdefiniowanych w bazie, do której jesteśmy aktualnie podłączeni, razem z ich rolą domyślną i grupą.

Przykład:

```
 UserName    GroupName  LoginName   UserID  SUID
------------ ---------- ----------- ------- -----
dbo          public     sa          1       1
adm_user     public     adm_user    2       5
```

```
 aplikacja    public     aplikacja   3       6


(3 rows affected)
```

TYLKO ODCZYT

Utworzenie loginu na poziomie serwera: sp_addlogin

```
sp_addlogin NazwaLoginu, 'TrudneHasło'
go
```

Zakładamy nowe konto logowania na poziomie całego serwera. Sam login nie daje jeszcze dostępu do żadnej konkretnej bazy — do tego potrzebny jest osobny krok `sp_adduser` w każdej bazie docelowej.

TYLKO ODCZYT

Utworzenie usera w bieżącej bazie danych: sp_adduser

```
sp_adduser NazwaLoginu
go
```

Wiążemy istniejący login z bieżącą bazą danych, tworząc dla niego usera — od tego momentu login może się do tej bazy podłączyć i operować w niej zgodnie z nadanymi dalej uprawnieniami.

TYLKO ODCZYT

Wylistowanie wszystkich ról: sp_displayroles

```
sp_displayroles
go
```

Zwracamy listę ról zdefiniowanych na serwerze — zarówno wbudowanych (`sa_role`, `sso_role`, `oper_role`), jak i utworzonych ręcznie.

Przykład:

```
 Role Name        Role ID
 ---------------- --------
 sa_role            0
 sso_role           1
 oper_role          2
 replication_role   3

(4 rows affected)
```

□□ TYLKO ODCZYT

Wylistowanie loginów zdefiniowanych na serwerze: sp_displaylogin

```
sp_displaylogin NazwaLoginu
go
```

Zwracamy szczegóły konkretnego loginu — jego status (aktywny/zablokowany), datę ostatniej zmiany hasła, przypisane role serwerowe i domyślną bazę danych.

Przykład:

```
Login name                      adm_user
Full name
Default database                produkcja
Default language                (Server default)
Auditing                        global default
Configured authorization to set system roles listed below
No system roles configured for adm_user

Locked                          NO
Password expiration interval    (Not set. Using server default of 0)
Failed login attempts           0
Login server assigned to         None
```

□□ ZMIENIA SYSTEM

Nadanie roli serwerowej użytkownikowi: grant role

```
grant role NazwaRoli to NazwaLoginu
go
```

Nadajemy loginowi jedną z ról serwerowych, na przykład `sa_role` albo `oper_role`, dającą mu szersze uprawnienia administracyjne niezależne od pojedynczej bazy danych.

ZMIENIA SYSTEM

Odebranie roli serwerowej: revoke role

```
revoke role NazwaRoli from NazwaLoginu
go
```

Odbieramy wcześniej nadaną rolę serwerową — login pozostaje aktywny, traci jedynie uprawnienia wynikające z tej konkretnej roli.

TYLKO ODCZYT

Zmiana hasła użytkownikowi: sp_password

```
sp_password 'StareHasło', 'NoweTrudneHasło', NazwaLoginu
go
```

Zmieniamy hasło wskazanego loginu — administrator z rolą `sso_role` może pominąć stare hasło i podać `NULL` w jego miejsce, żeby zresetować hasło bez znajomości poprzedniego.

Odpowiednik tej operacji w nowszej składni:

```
ALTER LOGIN NazwaLoginu WITH PASSWORD = 'NoweTrudneHasło'
go
```

ZMIENIA SYSTEM

Wymuszenie zmiany hasła przy najbliższym logowaniu

```
ALTER LOGIN NazwaLoginu WITH PASSWORD = 'NoweTrudneHasło' MUST_CHANGE
go
```

Ustawiamy hasło tymczasowe, które login musi zmienić przy pierwszym logowaniu po tej operacji — używamy tego przy wydawaniu nowych kont albo po podejrzeniu wycieku hasła.

□□ TYLKO ODCZYT

Usunięcie usera z bieżącej bazy danych: sp_dropuser

```
sp_dropuser NazwaLoginu
go
```

Usuwamy dostęp loginu do bieżącej bazy danych. Sam login na poziomie serwera pozostaje nietknięty i zachowuje dostęp do pozostałych baz, w których ma jeszcze usera.

□□ TYLKO ODCZYT

Usunięcie loginu z serwera: sp_droplogin

```
sp_droplogin NazwaLoginu
go
```

Usuwamy login z poziomu całego serwera. Operacja kończy się błędem, jeśli login ma jeszcze aktywnego usera w którejkolwiek bazie danych — najpierw wykonujemy `sp_dropuser` w każdej z nich.

Z życia wzięte

□□ Z ŻYCIA WZIĘTE

Zespół kadr dostaje nowego pracownika, który ma tylko odczytywać archiwum płac PLACE_ARCH do celów raportowych, bez prawa do żadnej zmiany. Zakładamy mu login przez sp_addlogin, a potem usera w bazie PLACE_ARCH przez sp_adduser, bez nadawania żadnej roli poza domyślnym dostępem do odczytu. Miesiąc później próbuje przypadkiem coś zaktualizować w danych archiwalnych i dostaje jasny błąd braku uprawnień, dokładnie tak jak powinno być.

Po co to istnieje

> **Pojęcie: login.** Login to konto istniejące na poziomie całego serwera Sybase, niezależnie od konkretnej bazy danych. Rejestr wszystkich loginów trzyma systemowa tabela `syslogins` w bazie `master`. Sam login nie daje jeszcze prawa do niczego poza uwierzytelnieniem się na serwerze — to punkt startowy, ale nie dostęp do danych.

> **Pojęcie: user.** User to konto na poziomie konkretnej bazy danych, powiązane z jednym loginem, przez które ten login faktycznie uzyskuje dostęp do obiektów tej bazy. Rejestr userów danej bazy trzyma tabela `sysusers`, obecna osobno w każdej bazie. Ten sam login może mieć różnych userów w różnych bazach (albo w ogóle nie mieć usera w jakiejś bazie, i wtedy nie ma do niej dostępu), co pozwala nadawać dostęp selektywnie, baza po bazie, jednym kontem serwerowym.

Kolejność zakładania konta jest więc zawsze taka sama: najpierw login na poziomie serwera, potem user w każdej bazie, do której ten login ma mieć dostęp.

Częste błędy

Usunięcie loginu przed usunięciem jego userów. `sp_droplogin` odmówi wykonania, dopóki w którejkolwiek bazie istnieje odpowiadający mu user — kolejność usuwania jest odwrotna do kolejności zakładania.

Założenie usera bez wcześniejszego loginu. `sp_adduser` wymaga istniejącego loginu na poziomie serwera; próba dodania usera dla nieistniejącego loginu kończy się błędem, mimo że nazwa wygląda identycznie jak przy zwykłej literówce.

Nadanie `sa_role` zamiast konkretnej roli aplikacyjnej. Rola `sa_role` daje pełne uprawnienia administracyjne na całym serwerze — nadawanie jej kontom aplikacyjnym łamie zasadę najmniejszych uprawnień i utrudnia później odtworzenie, kto faktycznie czego potrzebował.

Powiązane: isql: klient wiersza poleceń · Rozpoznanie serwera i baz · Konta, role i uprawnienia: GRANT, REVOKE, zasada najmniejszych uprawnień

Źródła: notatki własne administratora Sybase ASE · dokumentacja SAP ASE, rozdział „Security Administration Guide"

Revision #45
Created 2025-03-13 14:57:36 UTC by Tomasz Konieczny
Updated 2026-10-26 14:08:07 UTC by Tomasz Konieczny