psql: klient wiersza poleceń

Poziom: podstawy **Dotyczy: PostgreSQL 9.6-17** **Czas: ~20 min**

`psql` jest jedynym narzędziem, którego naprawdę trzeba się nauczyć w PostgreSQL — cała reszta administracji rozgrywa się wewnątrz tej samej sesji, metakomendami i zapytaniami do katalogu systemowego.

Skróty użyte na tej stronie (4)

Skrót	Co znaczy
CLI	Command Line Interface: praca przez wpisywanie komend w terminalu
CSV	w Windowsie: Cluster Shared Volume, dysk widoczny naraz dla wszystkich węzłów klastra; w danych: Comma-Separated Values, plik z wartościami oddzielonymi przecinkami
SQL	Structured Query Language: język zapytań do relacyjnych baz danych
TCP	Transmission Control Protocol: protokół sieciowy z potwierdzaniem dostarczenia

Fiszka

TYLKO ODCZYT

Wylistowanie wszystkich baz danych: \l

```
\l
```

Zwracamy listę baz danych w klastrze wraz z właścicielem, kodowaniem i uprawnieniami dostępu.

Przykład:

```
                                  List of databases
    Name    |  Owner   | Encoding |   Collate   |    Ctype    |   Access privileges
------------+----------+----------+-------------+-------------+-----------------------
 postgres   | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
```

```
 produkcja  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 template0  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres          +
            |          |          |             |             | postgres=CTc/postgres
 template1  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres          +
            |          |          |             |             | postgres=CTc/postgres
(4 rows)
```

Wariant z plusem, `\l+`, dokłada rozmiar każdej bazy i tablespace, w którym leży.

TYLKO ODCZYT

Wylistowanie wszystkich schematów: \dn

```
\dn
```

Zwracamy schematy istniejące w bieżącej bazie, do której jesteśmy podłączeni.

Przykład:

```
        List of schemas
    Name    |       Owner
------------+--------------------
 public     | postgres
 sprzedaz   | nazwa_usera
(2 rows)
```

TYLKO ODCZYT

Wylistowanie tabel z bieżącego schematu: \dt

```
\dt
```

Bez dodatkowego argumentu `\dt` pokazuje tabele wyłącznie z pierwszego schematu na ścieżce przeszukiwania (`search_path`) — zwykle `public`.

Przykład:

```
          List of relations
 Schema |    Name    | Type  |  Owner
```

```
--------+------------+-------+----------
 public | klienci    | table | postgres
 public | zamowienia | table | postgres
(2 rows)
```

◫◫ TYLKO ODCZYT

Wylistowanie tabel z konkretnego schematu: \dt NazwaSchematu.*

```
\dt sprzedaz.*
```

Wskazujemy schemat wprost, gdy interesują nas tabele spoza domyślnej ścieżki przeszukiwania.

Przykład:

```
             List of relations
  Schema  |     Name      | Type  |   Owner
----------+---------------+-------+------------
 sprzedaz | faktury       | table | nazwa_usera
 sprzedaz | pozycje_faktur| table | nazwa_usera
(2 rows)
```

◫◫ TYLKO ODCZYT

Wylistowanie tabel ze wszystkich schematów: \dt .

```
\dt *.*
```

Rozszerzamy wzorzec na wszystkie schematy w bazie, łącznie z `pg_catalog` i `information_schema`, jeśli nie odfiltrujemy ich ręcznie.

Przykład:

```
               List of relations
   Schema   |      Name       | Type  |   Owner
------------+-----------------+-------+------------
 public     | klienci         | table | postgres
 public     | zamowienia      | table | postgres
```

```
 sprzedaz   | faktury         | table | nazwa_usera
 sprzedaz   | pozycje_faktur  | table | nazwa_usera
(4 rows)
```

🔒 TYLKO ODCZYT

Wyświetlenie definicji tabeli: \d NazwaTabeli

```
\d zamowienia
```

> **Pojęcie: definicja obiektu w `\d`.** Ta metakomenda zwraca nie dane z tabeli, tylko jej strukturę — kolumny z typami, indeksy założone na tabeli, ograniczenia (`NOT NULL`, `FOREIGN KEY`, `CHECK`) i triggery, jeśli jakieś na tabeli wiszą.

Przykład:

```
                                  Table "public.zamowienia"
   Column    |  Type   | Collation | Nullable |                Default
-------------+---------+-----------+----------+---------------------------------------
 id          | integer |           | not null | nextval('zamowienia_id_seq'::regclass)
 klient_id   | integer |           | not null |
 data        | date    |           | not null |
 wartosc     | numeric |           |          |
Indexes:
    "zamowienia_pkey" PRIMARY KEY, btree (id)
    "idx_zamowienia_klient_id" btree (klient_id)
Foreign-key constraints:
    "zamowienia_klient_id_fkey" FOREIGN KEY (klient_id) REFERENCES klienci(id)
```

🔒 TYLKO ODCZYT

Wyświetlenie wszystkich tabel bieżącego schematu przez samo \d

```
\d
```

Bez argumentu `\d` działa identycznie jak `\dt`, ale dokłada do wyniku też widoki i sekwencje — pełny obraz obiektów relacyjnych schematu, nie tylko tabel.

Przykład:

```
               List of relations
 Schema |         Name          |   Type   |  Owner
--------+-----------------------+----------+----------
 public | klienci               | table    | postgres
 public | zamowienia            | table    | postgres
 public | zamowienia_id_seq     | sequence | postgres
(3 rows)
```

TYLKO ODCZYT

Wylistowanie wszystkich indeksów: \di

```
\di
```

Zwracamy indeksy niezależnie od tego, do której tabeli należą — przydatne przy szukaniu indeksu po nazwie, bez pamiętania, na jakiej tabeli leży.

Przykład:

```
                   List of relations
 Schema |           Name            | Type  |  Owner   |    Table
--------+---------------------------+-------+----------+-------------
 public | idx_zamowienia_klient_id  | index | postgres | zamowienia
 public | zamowienia_pkey           | index | postgres | zamowienia
 public | klienci_pkey              | index | postgres | klienci
(3 rows)
```

TYLKO ODCZYT

Wylistowanie wszystkich ról i użytkowników: \du

```
\du
```

W PostgreSQL role i użytkownicy to ten sam obiekt — różnicę i konsekwencje tego rozwija strona Użytkownicy, role i uprawnienia.

Przykład:

```
                                   List of roles
  Role name  |                         Attributes                         | Member of
-------------+------------------------------------------------------------+-----------
 nazwa_usera | Create DB                                                  | {}
 postgres    | Superuser, Create role, Create DB, Replication, Bypass RLS | {}
 zbx_monitor |                                                            |
{pg_monitor}
(3 rows)
```

Wariant z plusem, `\du+`, dokłada opis (`Description`) roli, jeśli został ustawiony komendą `COMMENT ON ROLE`.

◫◫ TYLKO ODCZYT

Wylistowanie wszystkich funkcji: \df

```
\df
```

Zwracamy funkcje i procedury widoczne na ścieżce przeszukiwania, wraz z typem zwracanym i listą argumentów.

Przykład:

```
                          List of functions
 Schema |     Name      | Result data type | Argument data types |  Type
--------+---------------+------------------+---------------------+--------
 public | licz_rabat    | numeric          | klient_id integer   | func
(1 row)
```

◫◫ TYLKO ODCZYT

Wylistowanie wszystkich widoków: \dv

```
\dv
```

Analogicznie do `\dt`, ale dla widoków — wariant z gwiazdką (`\dv *.*`) rozszerza wynik na wszystkie schematy.

Przykład:

```
              List of relations
 Schema |        Name         | Type |  Owner
--------+---------------------+------+----------
 public | v_zamowienia_klient  | view | postgres
(1 row)
```

🔒 TYLKO ODCZYT

Otwarcie edytora tekstowego dla bieżącego polecenia: \e

```
\e
```

Otwieramy w zewnętrznym edytorze (zmienna `EDITOR`, domyślnie `vi`) bufor z ostatnio wpisanym poleceniem — wygodne przy pisaniu dłuższych zapytań wielolinijkowych, których niewygodnie poprawia się bezpośrednio w terminalu. Po zapisaniu i zamknięciu edytora `psql` wykonuje zawartość bufora.

🔒 TYLKO ODCZYT

Edycja treści funkcji w edytorze: \ef

```
\ef NazwaFunkcji
```

Otwieramy w edytorze definicję istniejącej funkcji, gotową do modyfikacji — po zapisaniu `psql` sam wykonuje `CREATE OR REPLACE FUNCTION` z poprawioną treścią.

🔒 TYLKO ODCZYT

Przełączenie formatu wyniku na rozszerzony: \x

```
\x
```

> **Pojęcie: format rozszerzony.** Zamiast szerokiej tabeli z kolumnami obok siebie, każdy wiersz wyniku wypisywany jest pionowo, jedna kolumna pod drugą — nieocenione przy tabelach z wieloma kolumnami albo z długą treścią w jednej z nich, które w zwykłym formacie łamią się na ekranie.

Przykład:

```
\x
Expanded display is on.
```

```
SELECT * FROM zamowienia WHERE id = 1;
```

```
-[ RECORD 1 ]-------
id        | 1
klient_id | 42
data      | 2026-01-15
wartosc   | 1250.00
```

Wariant `\x auto` włącza format rozszerzony tylko wtedy, gdy wynik nie mieści się na szerokości terminala, i jest tym, co warto wpisać do `.psqlrc`.

TYLKO ODCZYT

Włączenie pomiaru czasu wykonania: \timing

```
\timing
```

Po włączeniu każde kolejne polecenie wypisuje na końcu czas, jaki zajęło jego wykonanie — pierwsza rzecz, którą włączamy przy diagnozowaniu wolnych zapytań opisanym na stronie Sesje, długie zapytania, kasowanie kwerendy.

Przykład:

```
Timing is on.
```

```
SELECT count(*) FROM zamowienia;
```

```
 count
-------
 48213
(1 row)

Time: 12.481 ms
```

TYLKO ODCZYT

Przełączenie się na inną bazę: \c

```
\c NazwaBazy
```

Zamykamy bieżące połączenie i otwieramy nowe, do innej bazy w tym samym klastrze — jedyny sposób w `psql`, żeby zobaczyć obiekty innej bazy, bo pojedyncze połączenie zawsze celuje w jedną konkretną bazę.

Przykład:

```
\c produkcja
```

```
You are now connected to database "produkcja" as user "nazwa_usera".
```

Metakomenda przyjmuje też host, port i użytkownika, dzięki czemu można nią też przełączyć się na inny serwer bez wychodzenia z `psql`:

```
\c produkcja nazwa_usera pg02 5432
```

◫◫ TYLKO ODCZYT

Wykonanie komendy powłoki bez wychodzenia z psql: \!

```
\! komenda
```

Uruchamiamy dowolną komendę systemową w podpowłoce, nie zamykając sesji `psql` — wygodne, żeby na przykład sprawdzić rozmiar pliku dumpa leżącego obok, bez przełączania terminali.

Przykład:

```
\! ls -lh /var/lib/pgsql/backup/
```

```
-rw-------. 1 postgres postgres 842M lut 12 03:00 produkcja_full.dump
```

◫◫ TYLKO ODCZYT

Wykonanie skryptu SQL z pliku: \i

```
\i SciezkaDoPliku
```

Wczytujemy i wykonujemy zawartość wskazanego pliku tak, jakby jego treść została wpisana ręcznie w sesji — standardowy sposób odtworzenia zrzutu w formacie `plain` z `pg_dump`, opisanego na stronie pg_dump, pg_dumpall, pg_restore.

Przykład:

```
\i /path/to/file_dump.sql
```

```
CREATE TABLE
COPY 48213
ALTER TABLE
```

SQL = Structured Query Language: język zapytań do relacyjnych baz danych

TYLKO ODCZYT

Import lub eksport danych przez klienta: \copy

```
\copy NazwaTabeli TO 'SciezkaDoPliku' WITH CSV HEADER
```

> **Pojęcie: `\copy` kontra `COPY`.** Polecenie SQL `COPY` czyta i zapisuje plik z perspektywy procesu serwera, więc plik musi leżeć na dysku serwera i być dostępny dla konta systemowego `postgres`. Metakomenda `\copy` robi to samo, ale plik czyta i zapisuje klient `psql` na swoim własnym dysku, przesyłając dane przez połączenie — dzięki temu eksport czy import działa też wtedy, gdy łączymy się z odległej maszyny i nie mamy dostępu do systemu plików serwera.

Przykład eksportu tabeli do pliku CSV:

```
\copy zamowienia TO '/home/nazwa_usera/zamowienia.csv' WITH CSV HEADER
```

```
COPY 48213
```

SQL = Structured Query Language: język zapytań do relacyjnych baz danych · **CSV** = w Windowsie: Cluster Shared Volume, dysk widoczny naraz dla wszystkich węzłów klastra; w danych: Comma-Separated Values, plik z wartościami oddzielonymi przecinkami

Z życia wzięte

Z ŻYCIA WZIĘTE

O 10:20 Ola prosi o dostęp do bazy cennik tylko na chwilę, żeby sprawdzić jedną tabelę. Łączymy się `psql -h pg01 -U ola.wisniewska cennik` i pokazujemy jej `\dt` zamiast dawać hasło do konta administracyjnego. Po skończonej robocie Ola wychodzi z sesji, a jej konto zostaje z uprawnieniami tylko do jednego schematu.

Czym jest psql

Pojęcie: psql. Interaktywny klient wiersza poleceń dostarczany razem z PostgreSQL, łączący w sobie zwykły interpreter SQL i zestaw własnych poleceń — metakomend zaczynających się od `\` — które nie są SQL-em, tylko skrótami wykonującymi za nas zapytania do katalogu systemowego albo zmieniającymi zachowanie samej sesji. Metakomenda `\dt` nie istnieje jako polecenie serwera — `psql` po jej wpisaniu sam buduje i wysyła odpowiednie `SELECT` do `pg_catalog`, a wynik formatuje w czytelną tabelę.

Parametry połączenia

Połączenie z instancją opisują cztery parametry, wszystkie opcjonalne — pominięte przyjmują wartości domyślne (host lokalny przez gniazdo uniksowe, port `5432`, użytkownik systemowy, baza o tej samej nazwie co użytkownik):

```
psql -h NazwaSerwera -p NumerPortu -U NazwaUsera -d NazwaBazy
```

Przykład połączenia do zdalnej instancji:

```
psql -h pg01 -p 5432 -U nazwa_usera -d produkcja
```

Po podaniu `-h` klient łączy się przez TCP, nawet jeśli serwer i klient siedzą na tym samym hoście — pominięcie `-h` wymusza gniazdo uniksowe, co jest tym, czego używa `sudo -u postgres psql` opisane na stronie Instalacja PostgreSQL z własnym katalogiem danych.

Plik .pgpass — hasło bez interaktywnego wpisywania

> **Pojęcie: .pgpass.** Plik w katalogu domowym użytkownika systemowego, trzymający hasła do połączeń — dzięki niemu skrypty i zadania `cron` łączą się z bazą bez interaktywnego pytania o hasło i bez wpisywania go wprost w treści skryptu. Każda linia to jeden wzorzec połączenia:

```
pg01:5432:produkcja:nazwa_usera:HasloUzytkownika
```

Gwiazdka w dowolnym polu dopasowuje każdą wartość. Plik musi mieć uprawnienia `0600` — z szerszymi uprawnieniami `libpq` go ignoruje, traktując jak nieistniejący.

Zmienne środowiskowe połączenia

Zamiast powtarzać flagi `-h`/`-p`/`-U`/`-d` w każdym wywołaniu, `psql` (a właściwie biblioteka `libpq`, z której korzysta też każde inne narzędzie CLI silnika, w tym `pg_dump`) czyta je ze zmiennych środowiskowych, jeśli flagi nie zostały podane:

Zmienna	Odpowiednik flagi
`PGHOST`	`-h`
`PGPORT`	`-p`
`PGUSER`	`-U`
`PGDATABASE`	`-d`
`PGPASSWORD`	hasło — odradzane, trafia do listy procesów; wolimy `.pgpass`

Plik .psqlrc — konfiguracja sesji

> **Pojęcie: .psqlrc.** Plik w katalogu domowym, wykonywany automatycznie przy każdym starcie `psql` — miejsce na metakomendy, które i tak wpisywalibyśmy ręcznie po każdym połączeniu, jak `\timing on` albo rozszerzony format wyniku. Plik `.psqlrc` przyjmuje dokładnie te same metakomendy, które opisuje reszta tej strony:

```
\timing on
\x auto
\set PROMPT1 '%n@%/%R%# '
```

Częste błędy

Podawanie hasła w PGPASSWORD w skrypcie widocznym dla innych userów. Zmienna środowiskowa procesu jest widoczna w `/proc/PID/environ` dla właściciela procesu i roota — `.pgpass` z uprawnieniami `0600` jest tu bezpieczniejszym rozwiązaniem.

Mylenie `\d` z `SELECT * FROM`. `\d NazwaTabeli` pokazuje strukturę, nie dane — nowa osoba czasem oczekuje zawartości tabeli i jest zaskoczona listą kolumn zamiast wierszy.

Powiązane: Podstawowe operacje na bazie i użytkownikach · postgresql.conf i pg_hba.conf · Sesje, długie zapytania, kasowanie kwerendy · pg_dump, pg_dumpall, pg_restore

Źródła: dokumentacja PostgreSQL, rozdział „psql" · `man 1 psql`

Revision #44
Created 2026-09-24 23:08:01 UTC by Tomasz Konieczny
Updated 2026-10-26 14:12:13 UTC by Tomasz Konieczny