ROLE

Tworzenie i usuwanie roli

Jako komenda sql

```
CREATE ROLE name;
DROP ROLE name;
```

W terminalu

```
createuser name
dropuser name
```

Tworzenie roli z logowaniem

```
CREATE ROLE name  LOGIN;
CREATE USER name;
```

`CREATE USER` jest równoważne z `CREATE ROLE`, z wyjątkiem tego że `CREATE USER` zawiera `LOGIN` domyślnie, podczas gdy `CREATE ROLE` nie.

Role

- login privilege

```
CREATE ROLE name LOGIN;
```

- superuser status

```
CREATE ROLE name SUPERUSER;
```

Superużytkownik bazy danych omija wszystkie kontrole uprawnień, z wyjątkiem prawa do logowania. Jest to niebezpieczny przywilej i nie należy go używać nieostrożnie; najlepiej jest wykonywać większość swojej pracy jako rola, która nie jest superużytkownikiem.

- database creation

```
CREATE ROLE name CREATEDB;
```

Rola musi mieć jawnie nadane uprawnienia do tworzenia baz danych (oprócz superużytkowników, ponieważ omijają oni wszystkie kontrole uprawnień).

- role creation

```
CREATE ROLE name CREATEROLE;
```

Rola musi mieć jawnie nadane uprawnienia do tworzenia kolejnych ról (oprócz superużytkowników, ponieważ omijają one wszystkie kontrole uprawnień). Aby utworzyć taką rolę, użyj . Rola z uprawnieniami może zmieniać i usuwać role, które zostały przyznane użytkownikowi za pomocą opcji.

- initiating replication

```
CREATE ROLE name REPLICATION LOGIN;
```

Rola musi mieć jawne uprawnienia do inicjowania replikacji strumieniowej (oprócz superużytkowników, ponieważ omijają oni wszystkie kontrole uprawnień).

- password

```
CREATE ROLE name PASSWORD 'string';
ALTER ROLE name WITH PASSWORD 'nowe_hasło';
```

Hasło jest istotne tylko wtedy, gdy metoda uwierzytelniania klienta wymaga od użytkownika podania hasła podczas łączenia się z bazą danych. Metody uwierzytelniania passwordi md5 wykorzystują hasła. Hasła bazy danych są oddzielne od haseł systemu operacyjnego.

- inheritance of privileges

```
CREATE ROLE name NOINHERIT;
```

Rola dziedziczy uprawnienia ról, których jest członkiem, domyślnie. Jednak aby utworzyć rolę, która nie dziedziczy uprawnień domyślnie,

- bypassing row-level security

```
CREATE ROLE name BYPASSRLS;
```

Rola musi mieć jawne pozwolenie na ominięcie każdej polityki zabezpieczeń na poziomie wiersza (RLS)

- connection limit

```
CREATE ROLE name CONNECTION LIMIT 'integer';
```

Limit połączeń może określać, ile równoczesnych połączeń może utworzyć rola. -1 (domyślnie) oznacza brak limitu.

Role predefiniowane

Rola	Dozwolony dostęp
pg_read_all_data	Odczyt wszystkich danych (tabele, widoki, sekwencje), jakby użytkownik miał prawa SELECT do tych obiektów oraz prawa USAGE do wszystkich schematów, nawet jeśli nie ma ich explicite. Ta rola nie ma atrybutu BYPASSRLS ustawionego. Jeśli RLS jest używany, administrator może ustawić BYPASSRLS na rolach, którym ta rola jest przydzielona.
pg_write_all_data	Zapis wszystkich danych (tabele, widoki, sekwencje), jakby użytkownik miał prawa INSERT, UPDATE i DELETE do tych obiektów oraz prawa USAGE do wszystkich schematów, nawet jeśli nie ma ich explicite. Ta rola nie ma atrybutu BYPASSRLS ustawionego. Jeśli RLS jest używany, administrator może ustawić BYPASSRLS na rolach, którym ta rola jest przydzielona.
pg_read_all_settings	Odczyt wszystkich zmiennych konfiguracyjnych, nawet tych, które normalnie są widoczne tylko dla superużytkowników.
pg_read_all_stats	Odczyt wszystkich widoków pg_stat_* i używanie różnych rozszerzeń związanych ze statystykami, nawet tych, które normalnie są widoczne tylko dla superużytkowników.
pg_stat_scan_tables	Wykonywanie funkcji monitorujących, które mogą blokować tabele na poziomie ACCESS SHARE, potencjalnie na długi czas.
pg_monitor	Odczyt/wykonywanie różnych widoków i funkcji monitorujących. Ta rola jest członkiem pg_read_all_settings, pg_read_all_stats i pg_stat_scan_tables.
pg_database_owner	Brak dostępu. Członkostwo składa się z właściciela bieżącej bazy danych.
pg_signal_backend	Wysyłanie sygnału do innego backendu w celu anulowania zapytania lub zakończenia jego sesji.
pg_read_server_files	Zezwala na odczyt plików z dowolnej lokalizacji, do której baza danych ma dostęp na serwerze, za pomocą funkcji COPY i innych funkcji dostępu do plików.

Rola	Dozwolony dostęp
pg_write_server_files	Zezwala na zapis do plików w dowolnej lokalizacji, do której baza danych ma dostęp na serwerze, za pomocą funkcji COPY i innych funkcji dostępu do plików.
pg_execute_server_program	Zezwala na uruchamianie programów na serwerze bazy danych jako użytkownik, pod którym działa baza danych, za pomocą funkcji COPY i innych funkcji umożliwiających uruchamianie programów po stronie serwera.
pg_checkpoint	Zezwala na wykonanie polecenia CHECKPOINT.
pg_maintain	Zezwala na wykonanie VACUUM, ANALYZE, CLUSTER, REFRESH MATERIALIZED VIEW, REINDEX oraz LOCK TABLE na wszystkich relacjach, jakby użytkownik miał prawa MAINTAIN do tych obiektów, nawet jeśli nie ma ich explicite.
pg_use_reserved_connections	Zezwala na używanie zarezerwowanych slotów połączeń poprzez reserved_connections.
pg_create_subscription	Zezwala użytkownikom z uprawnieniami CREATE na bazie danych na wykonanie polecenia CREATE SUBSCRIPTION.

Nadawanie roli predefiniowanej

```
CREATE ROLE nowy_uzytkownik WITH LOGIN PASSWORD 'haslo';
GRANT pg_monitor TO nowy_uzytkownik;
```

Uprawnienia nadawanie, usuwanie

stworzenie nowej roli ``sql CREATE ROLE nowy_uzytkownik WITH LOGIN PASSWORD 'haslo';

```
nadanie uprawnień roli predefiniowanych do roli
```sql
GRANT pg_monitor TO nowy_uzytkownik;
GRANT pg_monitor, pg_read_all_data TO nowy_uzytkownik;
```

Rodzaje uprawnień do roli

- na poziomie bazy danych
  - CONNECT – pozwala na połączenie z bazą danych.
  - CREATE – pozwala na tworzenie nowych obiektów (tabel, widoków itp.) w bazie danych.
  - TEMPORARY – pozwala na tworzenie tymczasowych obiektów w czasie trwania sesji.

```

```
GRANT CONNECT, CREATE ON DATABASE nazwa_bazy TO nowy_uzytkownik;
```

- na poziomie tabeli
  - SELECT – pozwala na odczyt danych z tabeli.
  - INSERT – pozwala na wstawianie nowych wierszy do tabeli.
  - UPDATE – pozwala na aktualizowanie istniejących wierszy w tabeli.
  - DELETE – pozwala na usuwanie wierszy z tabeli.
  - TRUNCATE – pozwala na usuwanie wszystkich wierszy z tabeli.
  - REFERENCES – pozwala na tworzenie kluczy obcych, które odnoszą się do tej tabeli.
  - TRIGGER – pozwala na tworzenie i zarządzanie triggerami dla tabeli.

```
GRANT SELECT, INSERT ON TABLE employees TO nowy_uzytkownik;
GRANT UPDATE ON TABLE employees TO nowy_uzytkownik;
```

- na poziomie widoku
  - SELECT – pozwala na odczyt danych z widoku.

```
GRANT SELECT ON VIEW employee_view TO nowy_uzytkownik;
```

- na poziomie funkcji
  - EXECUTE – pozwala na wykonywanie funkcji

```
GRANT EXECUTE ON FUNCTION my_function TO nowy_uzytkownik;
```

- na poziomie schematu
  - USAGE – pozwala na używanie obiektów w schemacie (np. wywoływanie funkcji, odwoływanie się do tabel).
  - CREATE – pozwala na tworzenie nowych obiektów w schemacie

```
GRANT USAGE ON SCHEMA public TO nowy_uzytkownik;
GRANT CREATE ON SCHEMA public TO nowy_uzytkownik;
```

Aby usunąć uprawnienia zamiast `GRANT` trzeba napisać `REVOKE`

```

```

```
REVOKE USAGE, CREATE ON SCHEMA public FROM nowy_uzytkownik;
```