

UPRAWNIENIA

Bezpieczeństwo

W PostgreSQL dostęp do danych jest przydzielany/odmawiany w zależności od roli/użytkownika.

W PostgreSQL występuje 6 poziomów bezpieczeństwa:

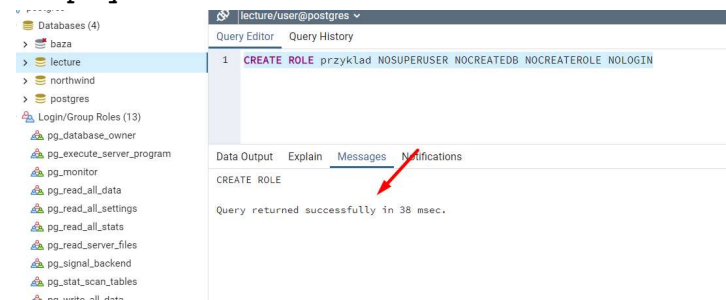
Bezpieczeństwo

Poziom	Opis
Instancji	Użytkownicy, role, tworzenie baz danych, logowanie, replikacja danych
Bazy danych	łączenie, tworzenie elementów w bazie
Schema	używanie i tworzenie obiektów w schemacie
Tabeli	Odczyt, wstawianie, aktualizacja
Kolumny	dostęp do danych z kolumny
Wiersza	dostęp do wierszy

Bezpieczeństwo

Poziom Instancji:

CREATE ROLE przykład NOSUPERUSER NOCREATEDB NOCREATEROLE NOLOGIN



The screenshot shows the 'Group Role - przykład' configuration window in MySQL. The 'Privileges' tab is selected, showing a list of permissions with toggle switches. The permissions and their states are:

Privilege	State
Can login?	Off
Superuser?	Off
Create roles?	Off
Create databases?	Off
Inherit rights from the parent roles?	On
Can initiate streaming replication and backups?	Off

At the bottom of the window, there are buttons for 'Close', 'Reset', and 'Save'. The 'Save' button is highlighted in blue.

Bezpieczeństwo

```
CREATE ROLE userno1 NOSUPERUSER NOCREATEDB NOCREATEROLE LOGIN  
PASSWORD 'xyz';  
  
CREATE USER userno2 NOSUPERUSER NOCREATEDB NOCREATEROLE LOGIN  
PASSWORD 'xyz2';
```

CREATE PASSW
CREATE PASSW

Login/Group Roles (16)

- pg_database_owner
- pg_execute_server_program
- pg_monitor
- pg_read_all_data
- pg_read_all_settings
- pg_read_all_stats
- pg_read_server_files
- pg_signal_backend
- pg_stat_scan_tables
- pg_write_all_data
- pg_write_server_files
- przyklad
- user
- user2
- userno1
- userno2
- Tablespaces

PRUSER NOCREATEDB NOCREATEROLE LOGIN

PRUSER NOCREATEDB NOCREATEROLE LOGIN

Bezpieczeństwo

Login Role - userno1

General Definition Privileges Membership Parameters Security SQL

Name: userno1

Comments:

Can login? ☒

Superuser? ☐

Create roles? ☐

Create databases? ☐

Inherit rights from the parent roles? ☒

Can initiate streaming replication and backups? ☐

Close Reset Save

Bezpieczeństwo

postgres_postgres_1 postgres
RUNNING PORT: 5432

CU

```
# psql -U userno1 -d lecture
psql (14.0 (Debian 14.0-1.pgdg110+1))
Type "help" for help.
```

Bezpieczeństwo

Użytkownik nie miał nadanych uprawnień dla bazy lecture, ale ze względu, na fakt, że jest ona w schemacie publicznym logowanie się udało.

Aby odebrać możliwość wykonywania operacji dla schematu publicznego:

```
REVOKE ALL ON DATABASE nazwa_bazy FROM public;
```

Bezpieczeństwo

Query Editor Query History

```
1 REVOKE ALL ON DATABASE lecture FROM public;
```

Data Output Explain Messages Notifications

REVOKE

Query returned successfully in 38 msec.

Bezpieczeństwo

Tym razem próba logowania:

postgres
RUNNING

postgres_admin_1 dpage/pgadmin4
RUNNING PORT: 8080

postgres_postgres_1 postgres
RUNNING PORT: 5432

Bezpieczeństwo

Poziom bazy danych:

Element	Opis
CREATE	Tworzenie nowego schematu
CONNECT	Łączenie z bazą danych
TEMP/TEMPORARY	Tworzenie tabel tymczasowych

Nadawanie uprawnień:

GRANT uprawnienie ON DATABASE nazwa_bazy TO nazwa_rolu/użytkownika

Bezpieczeństwo

Przykład:

Nadawania użytkownikowi uprawnienia logowania do bazy:

GRANT CONNECT ON DATABASE lecture TO userno1;

Query Editor Query History

```
1 GRANT CONNECT ON DATABASE lecture TO userno1;
2
```

Data Output Explain Messages Notifications

GRANT

Query returned successfully in 38 msec.

Bezpieczeństwo

Efekt:

```
# psql -U usern01 -d lecture
psql: error: connection to server on socket "/var/run/postgresql/
database "lecture"
DETAIL:  User does not have CONNECT privilege.
# psql -U usern01 -d lecture
psql (14.0 (Debian 14.0-1.pgdg110+1))
Type "help" for help.

lecture=>
```

Bezpieczeństwo

Poziom schematu:

Element	Opis
CREATE	Tworzenie tabel, funkcji, procedur, ...
USAGE	Odczyt schematu i jego obiektów

Bezpieczeństwo

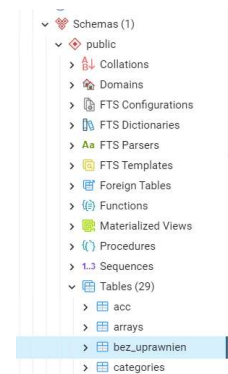
Czy użytkownik usern01 jest w stanie stworzyć tabelę w bazie danych lecture?

Bezpieczeństwo

Czy użytkownik usern01 jest w stanie stworzyć tabelę w bazie danych lecture?

```
# psql -U usern01 -d lecture
psql (14.0 (Debian 14.0-1.pgdg110+1))
Type "help" for help.

lecture=> CREATE TABLE bez_uprawnien(id int);
CREATE TABLE
```



Bezpieczeństwo

Podobnie jak poprzednio, problemem jest schemat publiczny.

Aby odebrać wszystkie uprawnienia dla schematu publicznego:

REVOKE ALL ON SCHEMA public FROM public;



Bezpieczeństwo

Aby nadać odpowiednie uprawnienia dla schematu w stosunku do użytkownika lub roli:

GRANT uprawnienia ON SCHEMA nazwa_schematu TO nazwa_rolu/użytkownika;

GRANT CREATE ON SCHEMA public TO userno1;

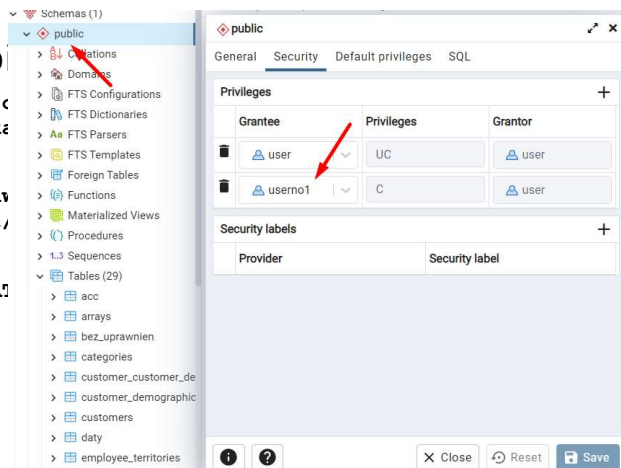


Bezpieczeństwo

Aby nadać odpowiednie uprawnienia dla schematu w stosunku do użytkownika lub roli:

GRANT uprawnienia ON SCHEMA nazwa_schematu TO nazwa_rolu/użytkownika;

GRANT CREATE ON SCHEMA public TO userno1;



Bezpieczeństwo

Poziom tabeli:

Element	Opis
SELECT	Odczytywanie danych z tabeli
INSERT	Wstawianie danych do tabeli
UPDATE	Aktualizacja danych w tabeli
DELETE	Usuwanie danych z tabeli
TRUNCATE	Usuwanie wszystkich danych w jednej operacji
TRIGGER	Tworzenie triggera dla tabeli
REFERENCES	Tworzenie ograniczenia klucza obcego

Bezpieczeństwo

Nadawanie uprawnień do wielu tabel jednocześnie:

```
GRANT uprawnienie ON ALL TABLES IN SCHEMA nazwa_schematu TO nazwa_rol;
```

Dla pojedynczej tabeli:

```
GRANT uprawnienie ON TABLE nazwa_tabeli TO nazwa_rol;
```

Bezpieczeństwo

Przykłady:

```
GRANT SELECT ON ALL TABLES IN SCHEMA public TO usernol;
```

```
GRANT INSERT ON TABLE daty TO usernol;
```

Bezpieczeństwo

Poziom kolumny:

Element	Opis
SELECT	Odczytywanie danych z kolumny
INSERT	Wstawianie danych do kolumny
UPDATE	Aktualizacja danych w kolumnie
REFERENCE	Tworzenie ograniczenia klucza obcego

```
GRANT uprawnienia (col1, col2, ...) ON nazwa_tabeli TO nazwa_rol;
```

Przykład - użytkownicy z uprawnieniem selekcji i wstawiania

```
<body>
<div class="app-wrapper">
  <input id='login' value='user'>
  <input id='password' value='password'>
  <button onclick="createConnection()">łącz</button>
  <button onclick="createTable()">Stwórz tabelę i nadaj uprawnienia</button>
  <button onclick="addData()">Dodaj dane</button>
  <button onclick="readData()">Odczytaj dane</button>
  <button onclick="endConnection()">Zakończ połączenie</button>
  <div id="div"></div>
</div>
```

Przykład - użytkownicy z uprawnieniem selekcji i wstawiania

```
<script>
const { Client } = require('pg');
let client = null;

function createConnection(){

  const login = document.getElementById('login').value;
  const password = document.getElementById('password').value;

  client = new Client({
    host: "127.0.0.1",
    port: 5432,
    database: 'perm',
    user: login,
    password: password
  });
  client.connect();
}
```

```
function createTable(){
const query = `
CREATE TABLE uprawnienia(
text varchar(100)
);
REVOKE ALL ON SCHEMA public FROM public;
CREATE USER no_permissions NOSUPERUSER NOCREATEDB NOCREATEROLE LOGIN PASSWORD
'AbcdEf123!';
CREATE USER read NOSUPERUSER NOCREATEDB NOCREATEROLE LOGIN PASSWORD 'AbcdEf123!';
CREATE USER read_write NOSUPERUSER NOCREATEDB NOCREATEROLE LOGIN PASSWORD 'AbcdEf123!';
REVOKE SELECT, INSERT, UPDATE, DELETE, TRUNCATE, TRIGGER, REFERENCES ON uprawnienia
FROM no_permissions;
REVOKE SELECT, INSERT, UPDATE, DELETE, TRUNCATE, TRIGGER, REFERENCES ON uprawnienia
FROM read;
REVOKE SELECT, INSERT, UPDATE, DELETE, TRUNCATE, TRIGGER, REFERENCES ON uprawnienia
FROM read_write;
GRANT SELECT ON uprawnienia TO read;
GRANT SELECT, INSERT ON uprawnienia TO read_write;
`;
client.query(query, (err, res)=>{
if(err){
console.log(err);
return;
}
console.log("Tabela stworzona, uprawnienia nadane");
}
```

Przykład - użytkownicy z uprawnieniem selekcji i wstawiania

```
function addData(){
const sql = "INSERT INTO uprawnienia(text) VALUES($1)";
const parameters = [Math.random()+'x'];
client.query(sql, parameters)
.then(res=>{
  console.log(res.rows[0]);
})
.catch(err=>console.log(err));
}
```

Przykład - użytkownicy z uprawnieniem selekcji i wstawiania

```
async function readData(){
const sql = "SELECT * FROM uprawnienia";
const query = {
  text: sql,
  rowMode: 'array',
};
const result = await client.query(query);
const div = document.querySelector("#div");
div.innerHTML = JSON.stringify(result.rows);
}

function endConnection(){
  client.end();
}
```