

[Przewodnik użytkownika](#) > PostgreSQL ustawianie uprawnień użytkownikowi

Od wersji 3.8 w systemie eDokumenty następuje sprawdzanie właściciela tabeli. W starszych wersjach był to *postgres* obecnie powinien być *edokumenty*.

Od wersji 6.0 w systemie Ready_™ pojawia się schema **epuap**. W przypadku czystej instalacji od systemu Ready_™ nie jest już dostępna schema **syncml_server**. Należy odpowiednio usunąć wpisy dotyczące schem z funkcji.

Przed przystąpieniem do aktualizacji systemu eDokumenty ze starszych wersji powinniśmy sprawdzić właściciela tabeli.

Wykonujemy to poleceniem:

```
\ds
```

Jeśli właścicielem tabeli, widoków są edokumenty nie musimy wykonywać żadnych dodatkowych czynności. W przeciwnym wypadku należy zmienić właściciela na *edokumenty* Aby to wykonać wykonujemy poniższe zapytania. Zapytania wykonujemy pojedynczo z uwagi gdyż przy hurtowym zapytaniu mogą się nie wykonać.

```
create or replace function my_exec1(text) returns void as $$
begin
execute $1;
end;
$$ language plpgsql;
```

```
SELECT my_exec1('ALTER SCHEMA ' || aa || ' OWNER TO edokumenty') FROM bs_unnest(array['public', 'epuap', 'syncml_server',
SELECT my_exec1('GRANT USAGE ON SCHEMA ' || aa || ' TO http') FROM bs_unnest(array['public', 'epuap', 'syncml_server', 'rep
```

```
SELECT my_exec1('ALTER TABLE ' || table_schema || '.' || table_name || ' OWNER TO edokumenty') FROM information_schema.views
SELECT my_exec1('GRANT SELECT ON ' || table_schema || '.' || table_name || ' TO http') FROM information_schema.views WHERE
```

```
select my_exec1('ALTER TABLE ' || schemaname || '.' || tablename || ' OWNER TO edokumenty') from pg_tables where schemaname
select my_exec1('GRANT select, insert, update, delete ON ' || schemaname || '.' || tablename || ' to http') from pg_tables
```

```
SELECT my_exec1('ALTER TABLE ' || sequence_schema || '.' || sequence_name || ' OWNER TO edokumenty') FROM information_sche
SELECT my_exec1('GRANT USAGE ON SEQUENCE ' || sequence_schema || '.' || sequence_name || ' TO http') FROM information_sche
```

```
SELECT my_exec1( 'ALTER FUNCTION ' || nsp.nspname || '.' || p.proname || '(' || pg_get_function_identity_arguments(p.oid) || ') OWNER
from pg_proc p
      join pg_namespace nsp ON p.pronamespace = nsp.oid
where nsp.nspname IN ('public', 'epuap', 'syncml_server', 'reports', 'vindication', 'cregisters', 'custom', 'jet_search',
SELECT my_exec1( 'GRANT EXECUTE ON FUNCTION ' || nsp.nspname || '.' || p.proname || '(' || pg_get_function_identity_arguments(p.oid)
from pg_proc p
      join pg_namespace nsp ON p.pronamespace = nsp.oid
where nsp.nspname IN ('public', 'epuap', 'syncml_server', 'reports', 'vindication', 'cregisters', 'custom', 'jet_search',
```

```
drop function my_exec1(text);
```

Dodawanie innego użytkownika do bazy i ustawianie uprawnień

Jeżeli chcemy ustawić uprawnienia do bazy dla danego użytkownika, to należy wykonać następujące operacje:

1. Dodać możliwość połączenia się do bazy dla użytkownika:

```
-- Usunięcie uprawnienia:
REVOKE CONNECT ON DATABASE edokumenty FROM PUBLIC;
-- Dodanie uprawnienia
GRANT CONNECT
ON DATABASE edokumenty TO uzytkownik;
```

1. Dodawanie uprawnień do dotychczas istniejących tabel/widoków: (należy być jako administrator / właściciel bazy podłączony do tej bazy)

```
-- Usuwanie uprawnień
REVOKE ALL
ON ALL TABLES IN SCHEMA public
FROM PUBLIC;

-- Dodawanie uprawnień
GRANT SELECT, INSERT, UPDATE, DELETE
ON ALL TABLES IN SCHEMA public
TO uzytkownik;
```

1. Ustawianie uprawnień dla wszystkich nowych obiektów (tabel/widoków):

```
ALTER DEFAULT PRIVILEGES
FOR ROLE uzytkownik -- alternatywnie "FOR USER"
IN SCHEMA public
GRANT SELECT, INSERT, UPDATE, DELETE ON TABLES TO uzytkownik;
```