

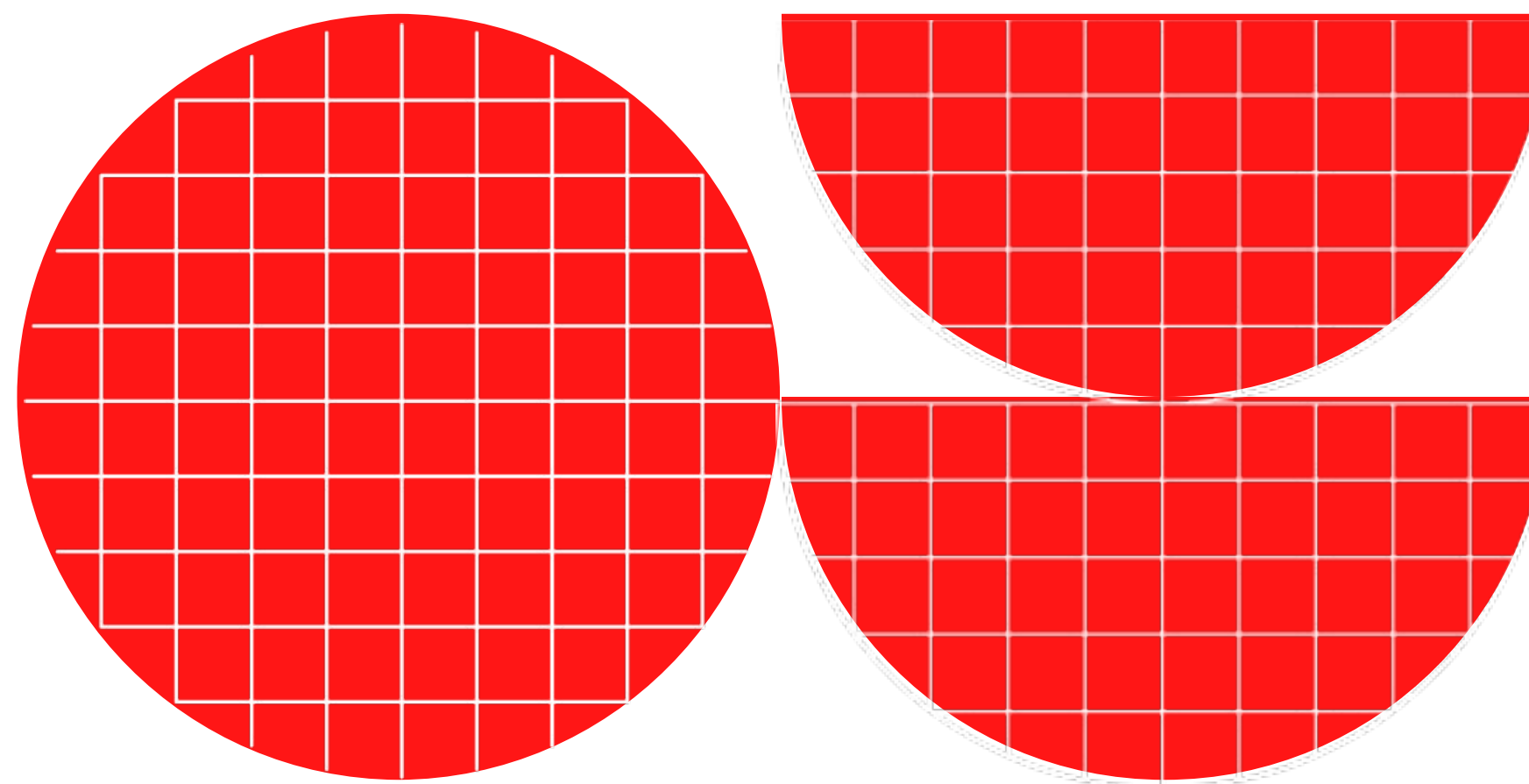


**CODE:ME**

UCZYMY PROGRAMOWANIA

# Dobro wraca ! SQL

Przemysław Wojtysiak



# Agenda

- Przedstawmy się
- Wprowadzenie do relacyjnych baz danych
- PostgreSQL
- PGAdmin4 - konfiguracja i logowanie
- Język SQL
- Pierwsza bazę
  - CREATE TABLE
  - INSERT
  - SELECT
  - UPDATE
  - DELETE
- Wnioski i dyskusja



# Przedstawmy się

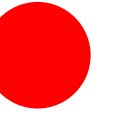
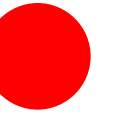


5 lat komercyjnego doświadczenie z PostgreSQL  
od 3 lat Microsoft SQL Server

Zawodowo administrator rozwiązań Contact Center

Pasjonat nowych technologii, krótkofalarstwa, podróży

# Co to jest baza danych?



***Zbiór danych zapisanych zgodnie z określonymi regułami. - Wikipedia***

***Zbiór informacji na jakiś temat przechowywanych i przetwarzanych przez komputer -***

*Słownik Języka Polskiego PWN*



# Elementy bazy danych - tabela

	customer_id [PK] integer	store_id smallint	first_name text	last_name text	email text	address_id smallint	activebool boolean	create_date date	last_update timestamp with time zone	active integer
1	1	1	MARY	SMITH	MARY.SMITH@sakilacustomer.org	5	true	2020-02-14	2020-02-15 10:57:20+01	1
2	2	1	PATRICIA	JOHNSON	PATRICIA.JOHNSON@sakilacustomer.org	6	true	2020-02-14	2020-02-15 10:57:20+01	1
3	3	1	LINDA	WILLIAMS	LINDA.WILLIAMS@sakilacustomer.org	7	true	2020-02-14	2020-02-15 10:57:20+01	1
4	4	2	BARBARA	JONES	BARBARA.JONES@sakilacustomer.org	8	true	2020-02-14	2020-02-15 10:57:20+01	1
5	5	1	ELIZABETH	BROWN	ELIZABETH.BROWN@sakilacustomer.org	9	true	2020-02-14	2020-02-15 10:57:20+01	1
6	6	2	JENNIFER	DAVIS	JENNIFER.DAVIS@sakilacustomer.org	10	true	2020-02-14	2020-02-15 10:57:20+01	1
7	7	1	MARIA	MILLER	MARIA.MILLER@sakilacustomer.org	11	true	2020-02-14	2020-02-15 10:57:20+01	1
8	8	2	SUSAN	WILSON	SUSAN.WILSON@sakilacustomer.org	12	true	2020-02-14	2020-02-15 10:57:20+01	1
9	9	2	MARGARET	MOORE	MARGARET.MOORE@sakilacustomer.org	13	true	2020-02-14	2020-02-15 10:57:20+01	1
10	10	1	DOROTHY	TAYLOR	DOROTHY.TAYLOR@sakilacustomer.org	14	true	2020-02-14	2020-02-15 10:57:20+01	1
11	11	2	LISA	ANDERSON	LISA.ANDERSON@sakilacustomer.org	15	true	2020-02-14	2020-02-15 10:57:20+01	1
12	12	1	NANCY	THOMAS	NANCY.THOMAS@sakilacustomer.org	16	true	2020-02-14	2020-02-15 10:57:20+01	1
13	13	2	KAREN	JACKSON	KAREN.JACKSON@sakilacustomer.org	17	true	2020-02-14	2020-02-15 10:57:20+01	1



# Elementy bazy danych - kolumna

	customer_id [PK] integer	store_id smallint	first_name text	last_name text	email text	address_id smallint	activebool boolean	create_date date	last_update timestamp with time zone	active integer
1	1	1	MARY	SMITH	MARY.SMITH@sakilacustomer.org	5	true	2020-02-14	2020-02-15 10:57:20+01	1
2	2	1	PATRICIA	JOHNSON	PATRICIA.JOHNSON@sakilacustomer.org	6	true	2020-02-14	2020-02-15 10:57:20+01	1
3	3	1	LINDA	WILLIAMS	LINDA.WILLIAMS@sakilacustomer.org	7	true	2020-02-14	2020-02-15 10:57:20+01	1
4	4	2	BARBARA	JONES	BARBARA.JONES@sakilacustomer.org	8	true	2020-02-14	2020-02-15 10:57:20+01	1
5	5	1	ELIZABETH	BROWN	ELIZABETH.BROWN@sakilacustomer.org	9	true	2020-02-14	2020-02-15 10:57:20+01	1
6	6	2	JENNIFER	DAVIS	JENNIFER.DAVIS@sakilacustomer.org	10	true	2020-02-14	2020-02-15 10:57:20+01	1
7	7	1	MARIA	MILLER	MARIA.MILLER@sakilacustomer.org	11	true	2020-02-14	2020-02-15 10:57:20+01	1
8	8	2	SUSAN	WILSON	SUSAN.WILSON@sakilacustomer.org	12	true	2020-02-14	2020-02-15 10:57:20+01	1
9	9	2	MARGARET	MOORE	MARGARET.MOORE@sakilacustomer.org	13	true	2020-02-14	2020-02-15 10:57:20+01	1
10	10	1	DOROTHY	TAYLOR	DOROTHY.TAYLOR@sakilacustomer.org	14	true	2020-02-14	2020-02-15 10:57:20+01	1
11	11	2	LISA	ANDERSON	LISA.ANDERSON@sakilacustomer.org	15	true	2020-02-14	2020-02-15 10:57:20+01	1
12	12	1	NANCY	THOMAS	NANCY.THOMAS@sakilacustomer.org	16	true	2020-02-14	2020-02-15 10:57:20+01	1
13	13	2	KAREN	JACKSON	KAREN.JACKSON@sakilacustomer.org	17	true	2020-02-14	2020-02-15 10:57:20+01	1



# Elementy bazy danych - wiersz

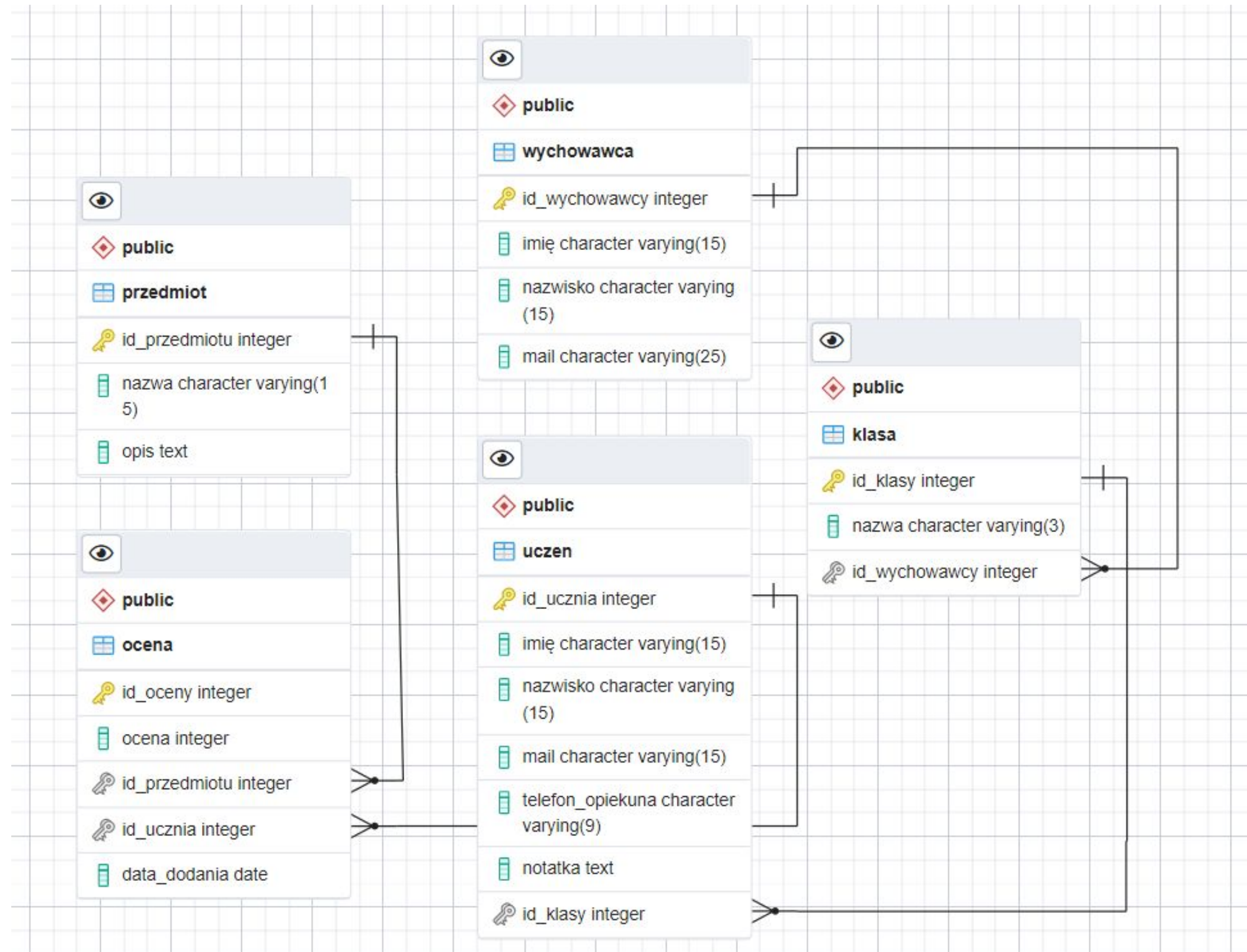
	customer_id [PK] integer	store_id smallint	first_name text	last_name text	email text	address_id smallint	activebool boolean	create_date date	last_update timestamp with time zone	active integer
1	1	1	MARY	SMITH	MARY.SMITH@sakilacustomer.org	5	true	2020-02-14	2020-02-15 10:57:20+01	1
2	2	1	PATRICIA	JOHNSON	PATRICIA.JOHNSON@sakilacustomer.org	6	true	2020-02-14	2020-02-15 10:57:20+01	1
3	3	1	LINDA	WILLIAMS	LINDA.WILLIAMS@sakilacustomer.org	7	true	2020-02-14	2020-02-15 10:57:20+01	1
4	4	2	BARBARA	JONES	BARBARA.JONES@sakilacustomer.org	8	true	2020-02-14	2020-02-15 10:57:20+01	1
5	5	1	ELIZABETH	BROWN	ELIZABETH.BROWN@sakilacustomer.org	9	true	2020-02-14	2020-02-15 10:57:20+01	1
6	6	2	JENNIFER	DAVIS	JENNIFER.DAVIS@sakilacustomer.org	10	true	2020-02-14	2020-02-15 10:57:20+01	1
7	7	1	MARIA	MILLER	MARIA.MILLER@sakilacustomer.org	11	true	2020-02-14	2020-02-15 10:57:20+01	1
8	8	2	SUSAN	WILSON	SUSAN.WILSON@sakilacustomer.org	12	true	2020-02-14	2020-02-15 10:57:20+01	1
9	9	2	MARGARET	MOORE	MARGARET.MOORE@sakilacustomer.org	13	true	2020-02-14	2020-02-15 10:57:20+01	1
10	10	1	DOROTHY	TAYLOR	DOROTHY.TAYLOR@sakilacustomer.org	14	true	2020-02-14	2020-02-15 10:57:20+01	1
11	11	2	LISA	ANDERSON	LISA.ANDERSON@sakilacustomer.org	15	true	2020-02-14	2020-02-15 10:57:20+01	1
12	12	1	NANCY	THOMAS	NANCY.THOMAS@sakilacustomer.org	16	true	2020-02-14	2020-02-15 10:57:20+01	1
13	13	2	KAREN	JACKSON	KAREN.JACKSON@sakilacustomer.org	17	true	2020-02-14	2020-02-15 10:57:20+01	1



# Elementy bazy danych - pole

	customer_id [PK] integer	store_id smallint	first_name text	last_name text	email text	address_id smallint	activebool boolean	create_date date	last_update timestamp with time zone	active integer
1	1	1	MARY	SMITH	MARY.SMITH@sakilacustomer.org	5	true	2020-02-14	2020-02-15 10:57:20+01	1
2	2	1	PATRICIA	JOHNSON	PATRICIA.JOHNSON@sakilacustomer.org	6	true	2020-02-14	2020-02-15 10:57:20+01	1
3	3	1	LINDA	WILLIAMS	LINDA.WILLIAMS@sakilacustomer.org	7	true	2020-02-14	2020-02-15 10:57:20+01	1
4	4	2	BARBARA	JONES	BARBARA.JONES@sakilacustomer.org	8	true	2020-02-14	2020-02-15 10:57:20+01	1
5	5	1	ELIZABETH	BROWN	ELIZABETH.BROWN@sakilacustomer.org	9	true	2020-02-14	2020-02-15 10:57:20+01	1
6	6	2	JENNIFER	DAVIS	JENNIFER.DAVIS@sakilacustomer.org	10	true	2020-02-14	2020-02-15 10:57:20+01	1
7	7	1	MARIA	MILLER	MARIA.MILLER@sakilacustomer.org	11	true	2020-02-14	2020-02-15 10:57:20+01	1
8	8	2	SUSAN	WILSON	SUSAN.WILSON@sakilacustomer.org	12	true	2020-02-14	2020-02-15 10:57:20+01	1
9	9	2	MARGARET	MOORE	MARGARET.MOORE@sakilacustomer.org	13	true	2020-02-14	2020-02-15 10:57:20+01	1
10	10	1	DOROTHY	TAYLOR	DOROTHY.TAYLOR@sakilacustomer.org	14	true	2020-02-14	2020-02-15 10:57:20+01	1
11	11	2	LISA	ANDERSON	LISA.ANDERSON@sakilacustomer.org	15	true	2020-02-14	2020-02-15 10:57:20+01	1
12	12	1	NANCY	THOMAS	NANCY.THOMAS@sakilacustomer.org	16	true	2020-02-14	2020-02-15 10:57:20+01	1
13	13	2	KAREN	JACKSON	KAREN.JACKSON@sakilacustomer.org	17	true	2020-02-14	2020-02-15 10:57:20+01	1

# Relacyjna baza danych







# PostgreSQL

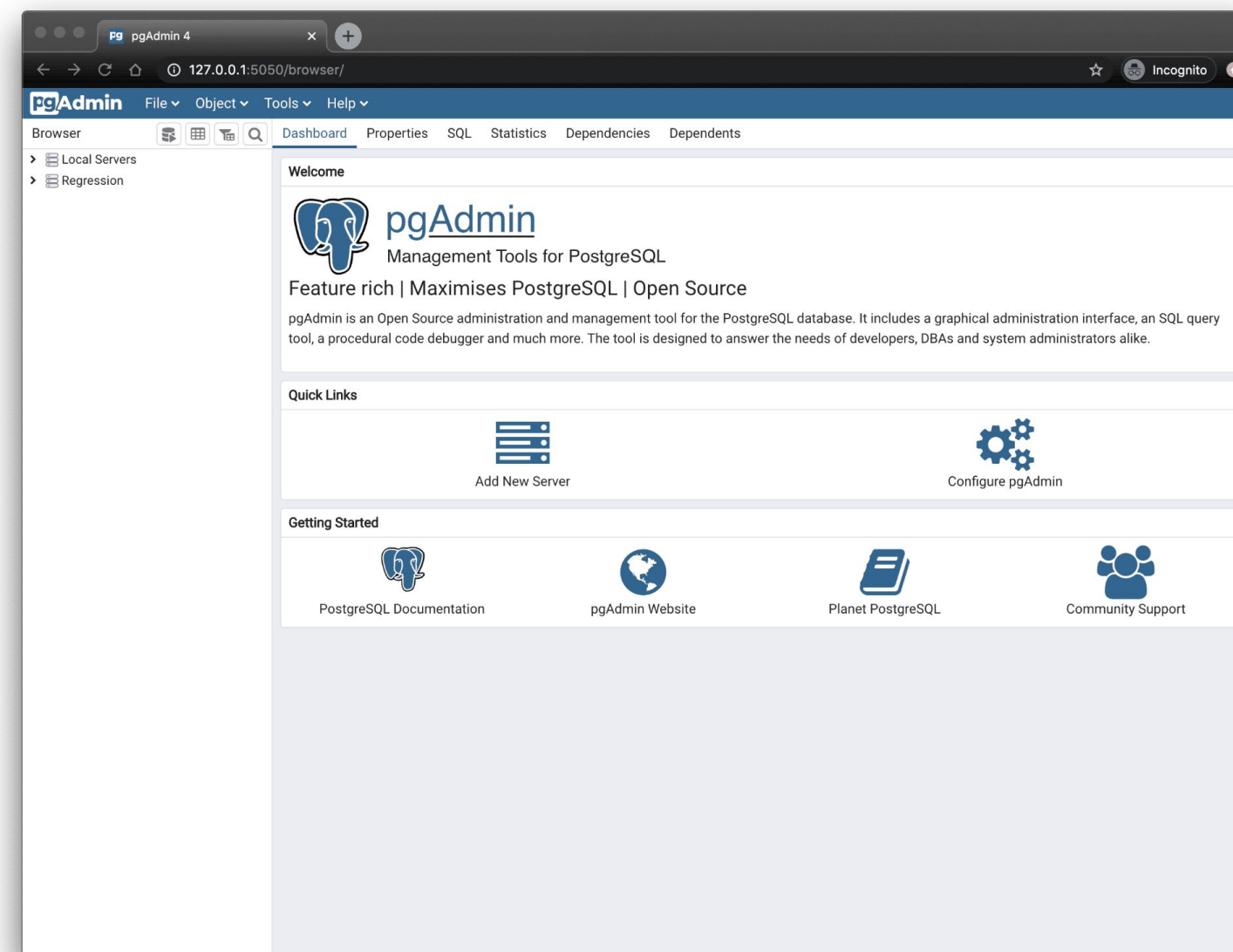
- Jeden z najpopularniejszych systemów RDBMS (system zarządzania relacyjną bazą danych)
- **Licencja** PostgreSQL (oparta na założeniach OpenSource)
- Wyjątkowo stabilne rozwiązanie
- Multiplatformowość
- Rozwijany od 1995 roku - **ostatni update 02.2022** - wersja 14.1
- Dostępny w jako Platform as a Service (PaaS) m.in Microsoft Azure, Google Cloud Platform
- Wykorzystywany przez wiodące firmy na świecie:





# PGAdmin4

- Narzędzie dla administratorów i deweloperów ułatwiające pracę z bazą PostgreSQL
- Opensource
- Multiplatformowość
- Wygodna alternatywa dla CLI
- Wygodne kreatory np. zamiast CREATE



# Konta użytkowników

- **Hostname/address (adres serwera) :**
- **Port:**
- **Username:** `codeme_student{id}` np. **codeme\_student001**
- **Password (hasło):** `codeme{id}` np. **codeme001**
- **Maintenance database (baza):** `codeme{id}` np **codeme001**

# PGAdmin4 - konfiguracja

- Ustawiamy hasło master PGAdmin
- **Hasło jest wymagane do ustawienia**

**Set Master Password**

Please set a master password for pgAdmin.  
This will be used to secure and later unlock saved passwords and other credentials.

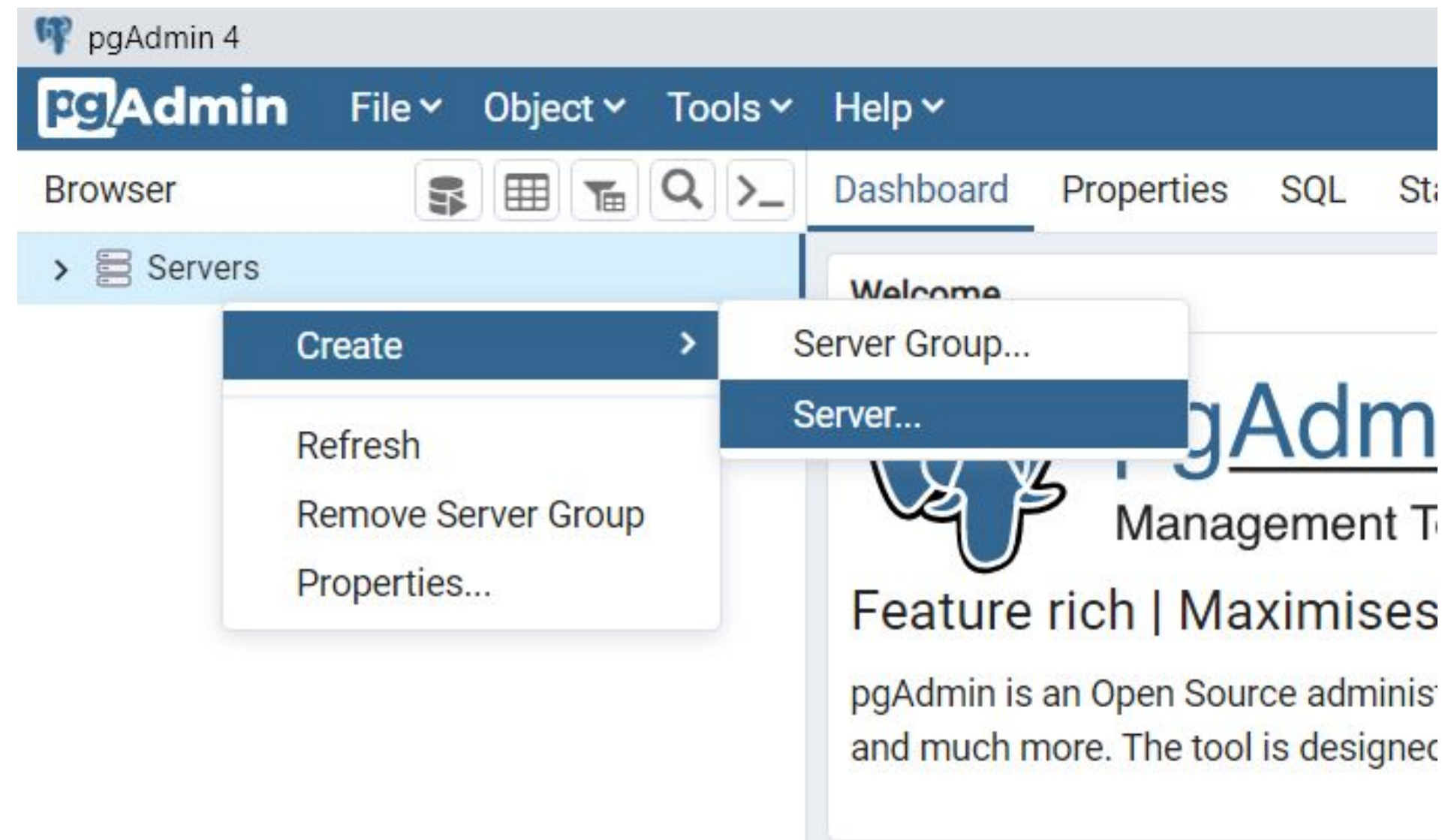
Password

?✕ Cancel✓ OK



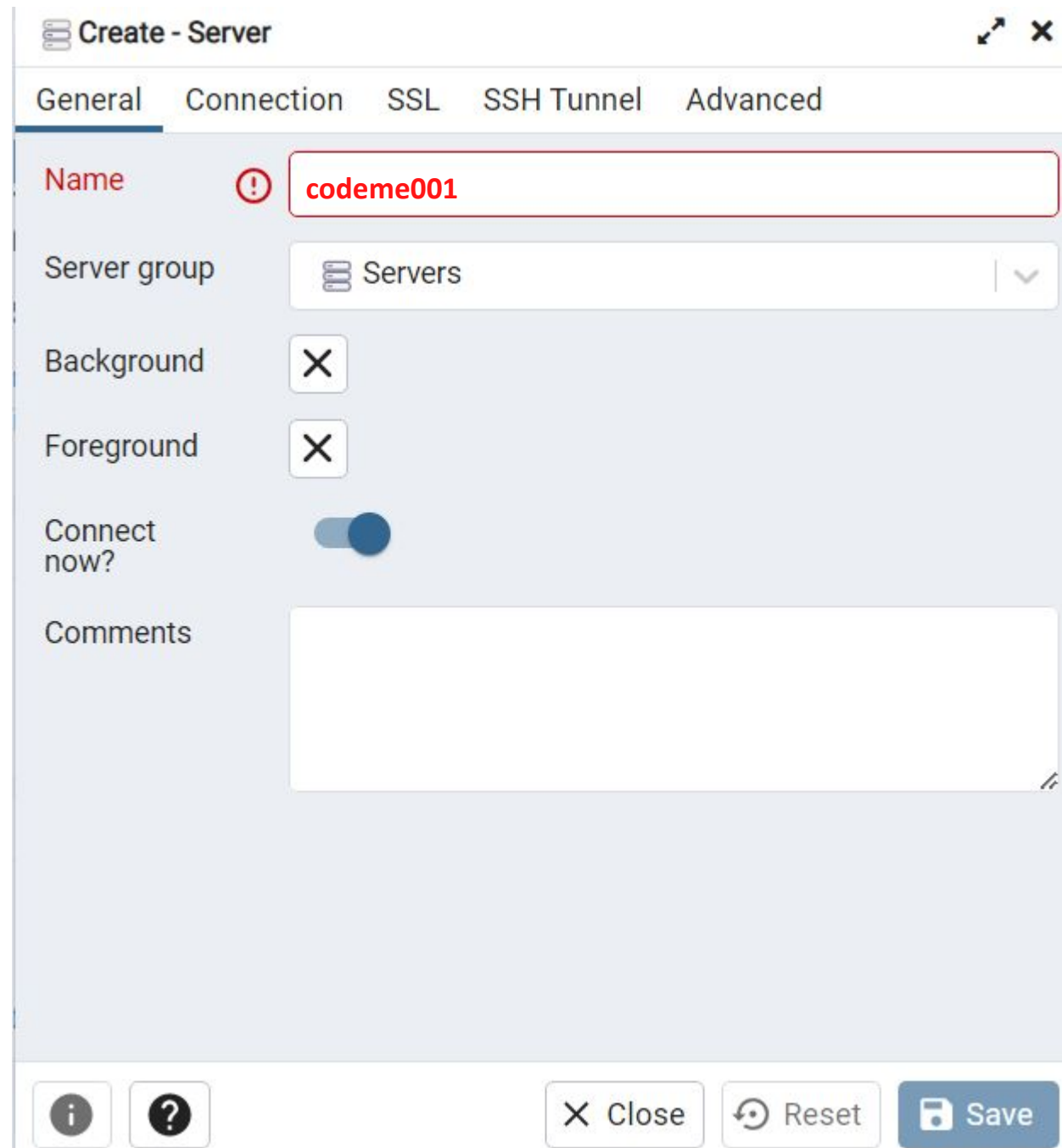
# PGAdmin4 - konfiguracja

- Dodanie nowego serwera PostgreSQL i bazy
- **Object > Create > Server**



# PGAdmin4 - logowanie

- **Object > Create > Server**



The 'Create - Server' dialog box in PGAdmin4, General tab. The 'Name' field contains 'codeme001' with a red warning icon. The 'Server group' is set to 'Servers'. 'Background' and 'Foreground' checkboxes are unchecked. 'Connect now?' is checked. The 'Comments' field is empty. At the bottom are buttons for 'Close', 'Reset', and 'Save'.

Create - Server

General Connection SSL SSH Tunnel Advanced

Name ! codeme001

Server group Servers

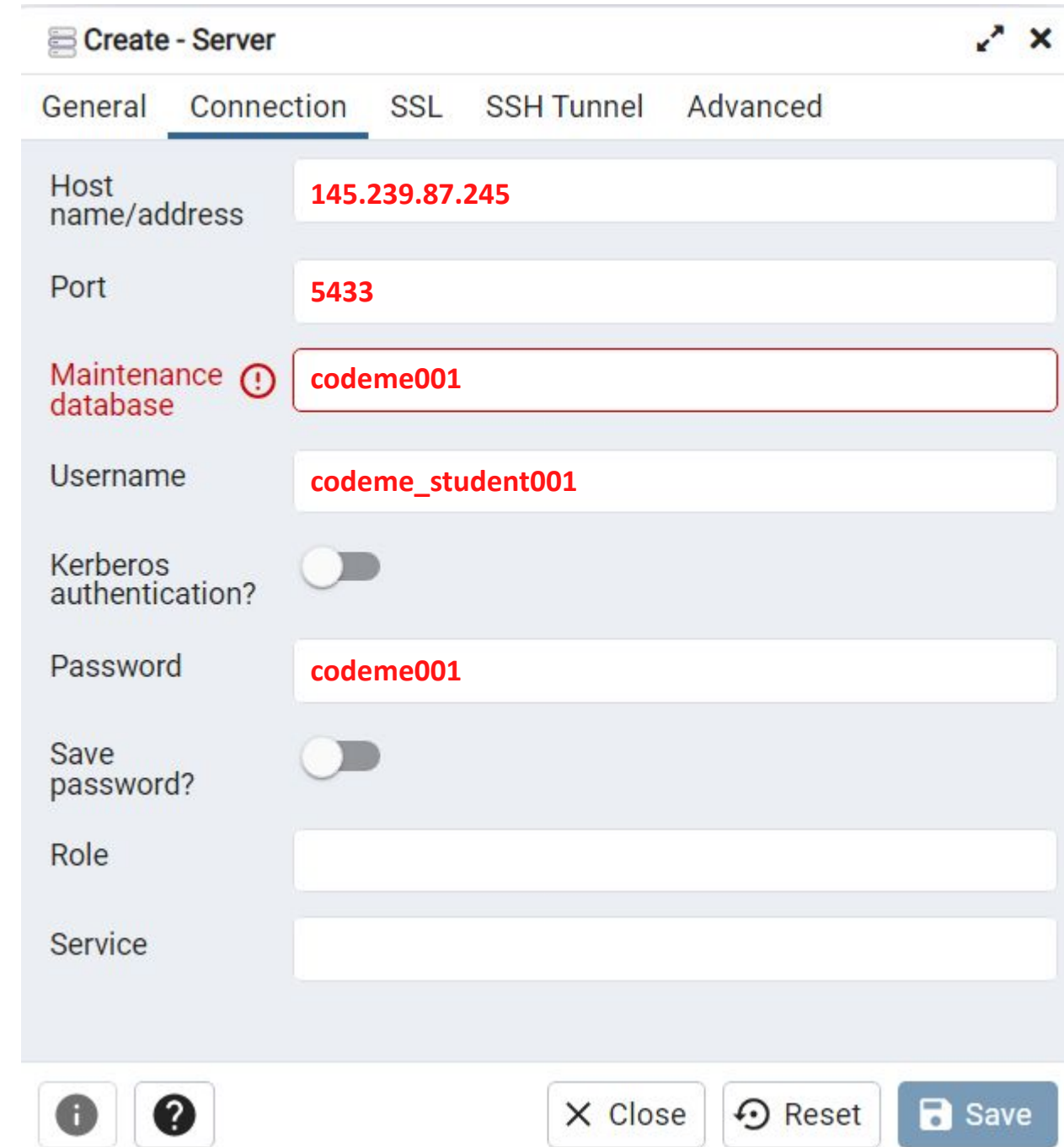
Background ☐

Foreground ☐

Connect now? ☒

Comments

Close Reset Save



The 'Create - Server' dialog box in PGAdmin4, Connection tab. 'Host name/address' is '145.239.87.245'. 'Port' is '5433'. 'Maintenance database' is 'codeme001' with a red warning icon. 'Username' is 'codeme\_student001'. 'Kerberos authentication?' and 'Save password?' are unchecked. 'Role' and 'Service' fields are empty. At the bottom are buttons for 'Close', 'Reset', and 'Save'.

Create - Server

General Connection SSL SSH Tunnel Advanced

Host name/address 145.239.87.245

Port 5433

Maintenance database ! codeme001

Username codeme\_student001

Kerberos authentication? ☐

Password codeme001

Save password? ☐

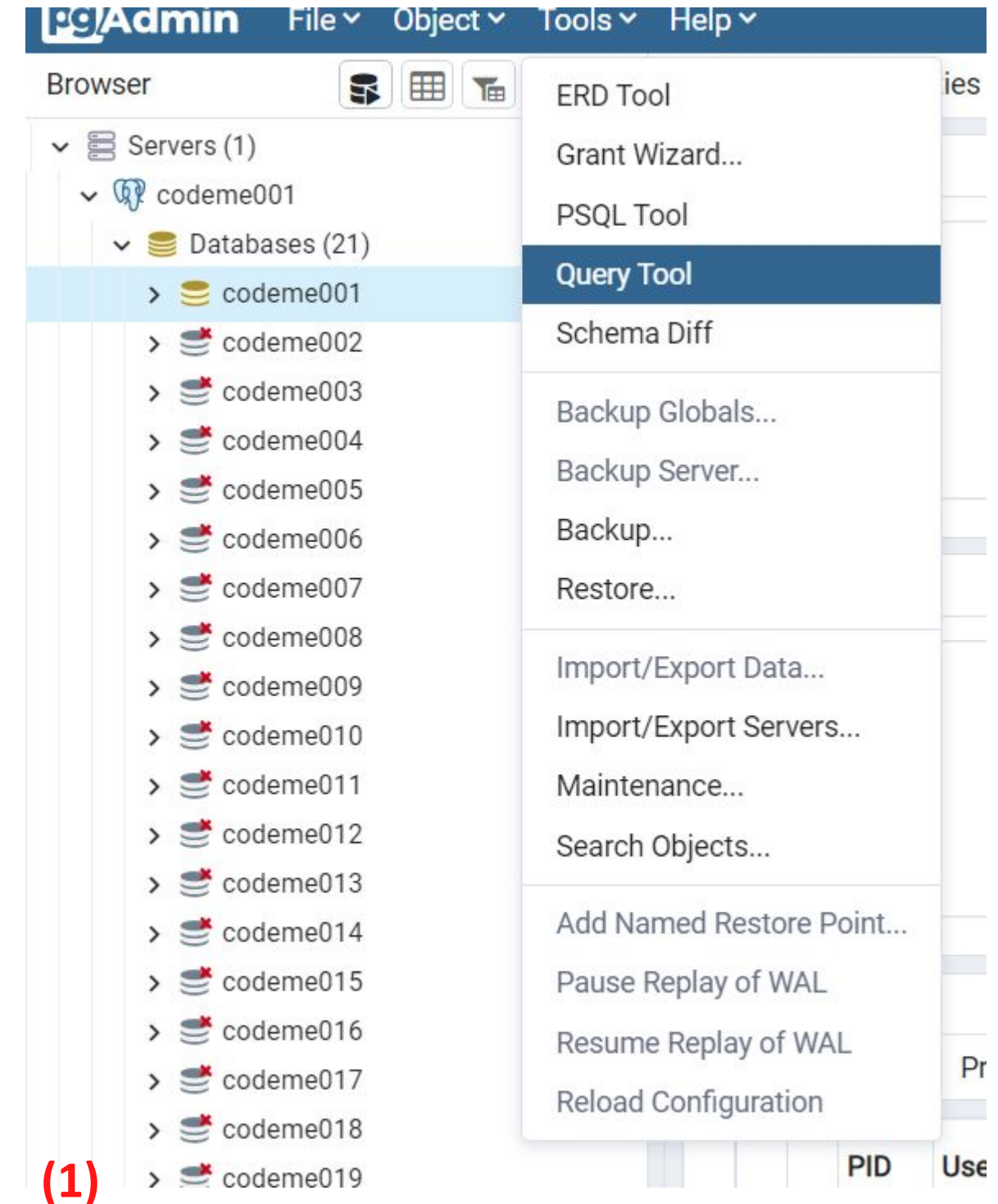
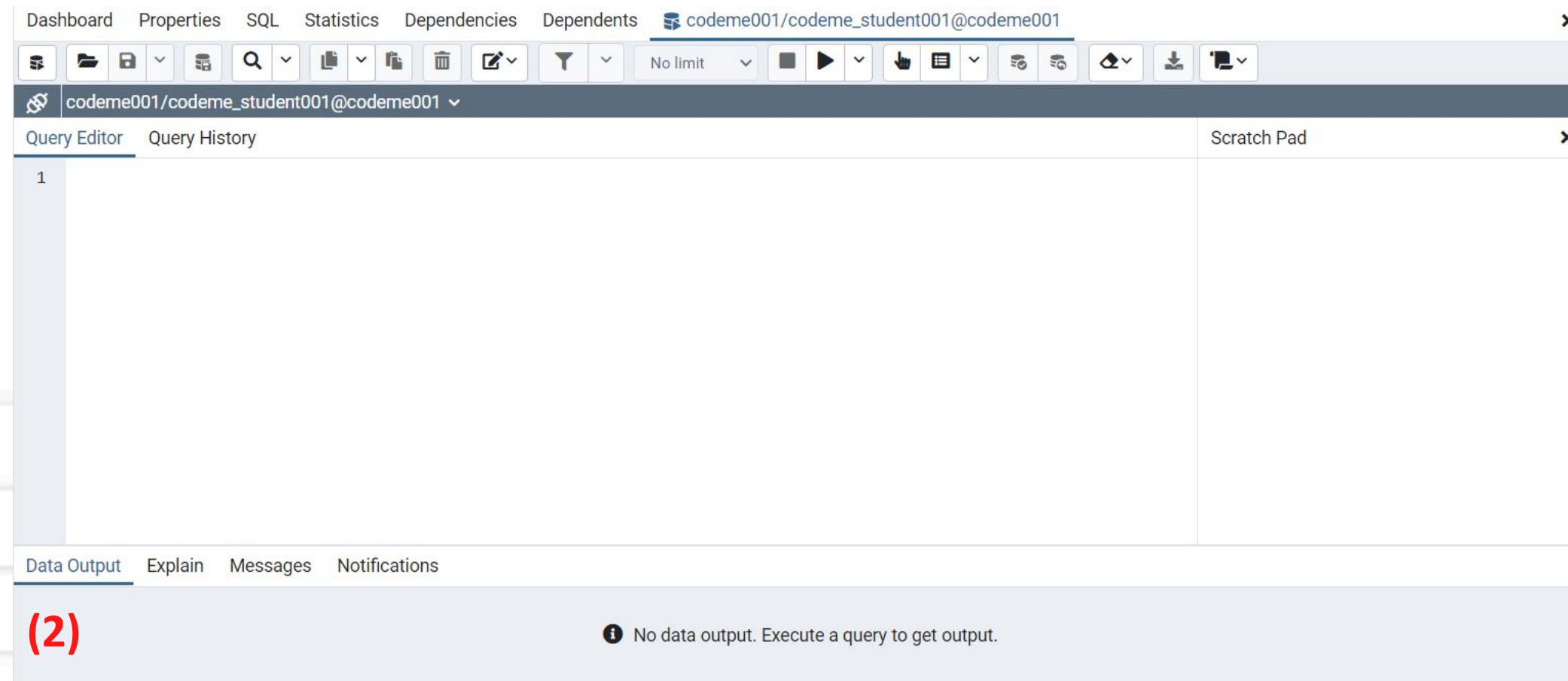
Role

Service

Close Reset Save

# PGAdmin4 - logowanie

- Wybieramy bazę (1)  
**Servers > codeme001 > Database > codeme001**
- Uruchamiamy okno poleceń SQL (Query Tools) (2)  
**Tools > Query Tools**





# SQL



Strukturalny język zapytań używany do tworzenia, modyfikowania baz danych oraz do umieszczania oraz modyfikacji danych w tych bazach.

## **Zasady:**

1. Polecenia mogą być rozmieszczone w kilku liniach
2. Koniec polecenia zaznacza się średnikiem ;
3. Można (zaleca się) używać tabulacji
4. Słowa nie mogą być dzielone
5. Wielkość liter nie ma znaczenia dla samego SQL (zalecane użycie dużych liter dla słów kluczowych SQL)

# SQL - przykłady

-zestawienie klienta

```
SELECT nazwa, opis
```

```
FROM przedmioty
```

```
WHERE nazwa='biologia';
```

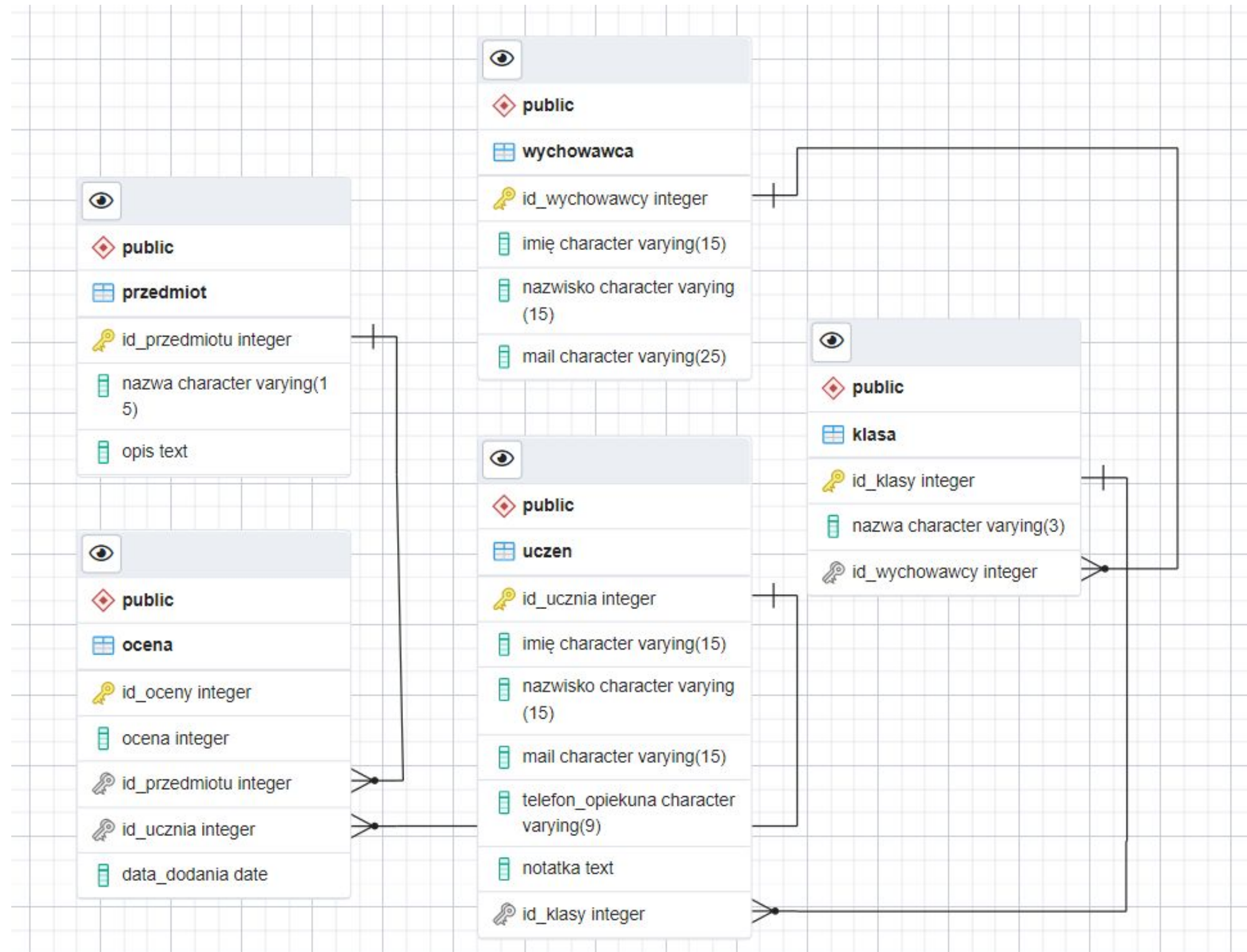
-aktualizacja opisu przedmiotu

```
UPDATE przedmioty
```

```
SET opis='Biologia poziom zaawansowany'
```

```
WHERE przedmiot='biologia';
```

# Baza danych - szkoła





# SQL: CREATE TABLE

```
CREATE TABLE table_name (  
    column1 datatype(length) column_constraint,  
    column2 datatype(length) column_constraint,  
    column3 datatype(length) column_constraint,  
    table_constraints  
);
```

- **Podstawowe typy danych:** *BOOL, VARCHAR, TEXT, SERIAL, INT, DOUBLE, DATE*
- **Podstawowe ograniczenia:** *NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY*

*Dobra praktyka to używanie nazwy tabel w formie pojedynczej.*

<https://www.postgresql.org/docs/current/sql-createtable.html>

# SQL: CREATE TABLE

```
CREATE TABLE przedmiot(  
    id_przedmiotu SERIAL PRIMARY KEY,  
    nazwa VARCHAR(15) NOT NULL,  
    opis TEXT  
);
```

<https://www.postgresql.org/docs/current/sql-createtable.html>

# SQL: INSERT

```
INSERT INTO table_name  
VALUES (value1, value2, value3, ...)  
RETURNING *
```

```
INSERT INTO table_name (column1, column2, column3, ...)  
VALUES  
    (value1, value2, value3, ...),  
    (value1, value2, value3, ...),  
    ...  
    (value1, value2, value3, ...);
```

- Brak określenia kolumn, wymaga uzupełnienia wszystkich wartości
- Pola typu **SERIAL** można pominąć, zostanie automatycznie wypełnione kolejnym indeksem.
- Pola z bez ograniczania **NOT NULL**, również mogą zostać pominięte wówczas przyjmą wartość Null
- **RETURNING \*** zwróci wszystkie wiersze, których dotyczył INSERT

<https://www.postgresql.org/docs/current/sql-insert.html>

# SQL: INSERT

```
INSERT INTO przedmiot  
VALUES (DEFAULT, 'informatyka', 'Programowanie i sieci');
```

```
INSERT INTO przedmiot (nazwa)  
VALUES ('geografia');
```

```
INSERT INTO przedmiot (nazwa, opis)  
VALUES ('informatyka2', 'Zaawansowane programowanie');
```

<https://www.postgresql.org/docs/current/sql-insert.html>



# SQL: INSERT - ćwiczenia

1. Dodaj nowego wychowawcę **Agata Wawrzyniak**
2. Dodaj klasę **CM**, której wychowawcą będzie wcześniej utworzony wychowawca
3. Dodaj uczniów **Iwona Nowak** i **Jadwiga Ratajska** do klasy **CM**



# SQL: SELECT

```
SELECT * FROM table;
```

```
SELECT column1, column2, column3  
FROM table;
```

```
SELECT * FROM table;  
WHERE column1 = "yz";
```

```
SELECT * FROM table LIMIT 100;
```

```
SELECT last_name, salary, 12*salary FROM employees;
```

- **Najważniejsze polecenie języka SQL !**
- \* zastępuje wszystkie kolumny - dobra praktyka to unikanie \*
- Warto określić **LIMIT** który ograniczy dane
- Wewnątrz SELECT możemy wykonywać różne operacje np. arytmetycznie
- Do łączenia tabel wykorzystujemy **JOIN**
- Możliwe wykonywanie operacji np. arytmetycznych

<https://www.postgresql.org/docs/current/sql-select.html>

# Funkcje agregujące

- AVG( ) - średnia arytmetyczna
- MAX( ) - wartość maksymalna
- MIN( ) - wartość minimalna
- SUM( ) - suma
- COUNT( ) - zliczanie

```
SELECT COUNT(*) FROM przedmiot;
```

<https://www.postgresql.org/docs/current/sql-select.html>

# SQL: SELECT

```
SELECT AVG(ocena)::numeric(3) AS Średnia  
FROM ocena  
WHERE id_przedmiotu=1
```

```
SELECT nazwa, imię, nazwisko  
FROM klasa  
LEFT JOIN wychowawca ON wychowawca.id_wychowawcy=klasa.id_wychowawcy
```

```
SELECT *  
FROM uczniowie  
LEFT JOIN klasa ON klasa.id_klasa=uczniowie.id_klasy  
WHERE klasa.nazwa='1A'
```

<https://www.postgresql.org/docs/current/sql-select.html>



# SQL: SELECT - ćwiczenia

1. Wyświetl wszystkich uczniów którzy nazywają się **Kowalski**
2. Policz ile jest klas w szkole
3. Policz ilu uczniów należy do klasy 1A



# SQL: UPDATE

```
UPDATE table_name  
SET column1 = value1,  
    column2 = value2,  
    ...  
WHERE condition;
```

- **Polecenie numer 2 języka SQL !**
- Modyfikacja istniejących danych
- Koniecznie zwracajmy uwagę na WHERE

<https://www.postgresql.org/docs/current/sql-update.html>

# SQL: UPDATE

```
UPDATE wychowawca  
SET nazwisko='Wolska'  
WHERE imię='Iwona' and nazwisko='Kowalczyk'
```

<https://www.postgresql.org/docs/current/sql-update.html>

# SQL: UPDATE - ćwiczenia

1. Zmień nazwisko wychowawcy **Agata Warzyniak** na **Agata Nowak**
2. Co się stanie po wykonaniu takiego kodu:

```
UPDATE przedmioty SET opis = 'Opis przedmiotu';
```





# SQL: DELETE

```
DELETE FROM table_name  
WHERE condition;
```

```
DELETE FROM uczen  
WHERE id_ucznia in (1,2,3);
```

- **Polecenie numer 3 języka SQL !**
- Koniecznie pamiętajmy o WHERE
- **DELETE** można 'mieszać' z **SELECT**

<https://www.postgresql.org/docs/14/sql-delete.html>

# SQL: DELETE - ćwiczenia

1. Usuń ucznia **Zbigniew Janik**
2. Zweryfikuj, co stało się z ocenami tego ucznia (do wykorzystania polecenie SELECT)



# Pagila - port Sakila z MySQL

- Testowa/demonstracyjna baza danych przygotowana przez zespół MySQL
- Doskonała do nauki zapytań SELECT
- Licencja OpenSource

<https://dev.mysql.com/doc/sakila/en/sakila-structure.html>

# Kody

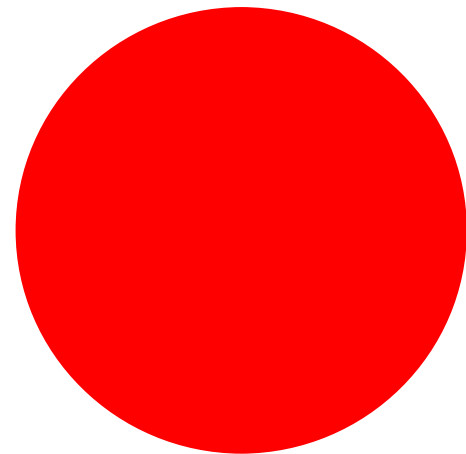
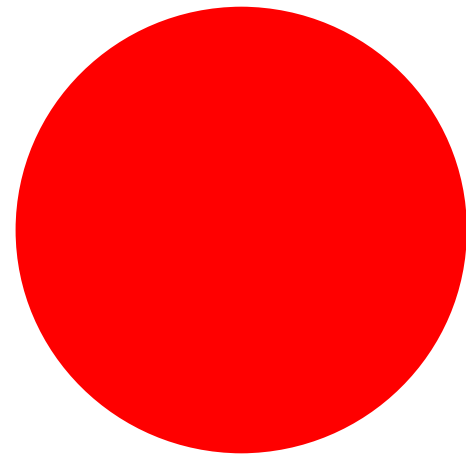
- CREATE TABLE: <https://pastebin.com/raw/HKaBus5a>
  - INSERT: <https://pastebin.com/raw/yDtJJpX1>
  - SELECT: <https://pastebin.com/ngbZ2AGN>
  - UPDATE: <https://pastebin.com/raw/H0aFfAsZ>
  - DELETE: <https://pastebin.com/raw/hNAyGA4a>
- 
- ĆWICZENIA INSERT: <https://pastebin.com/raw/UV4hqiRE>
  - ĆWICZENIA SELECT: <https://pastebin.com/raw/vViSYrTL>
  - ĆWICZENIA UPDATE: <https://pastebin.com/raw/qNQbSFjE>
  - ĆWICZENIA DELETE: <https://pastebin.com/raw/GE14aBth>



# Linki

- <https://www.postgresql.org/>
- <https://www.postgresql.org/docs/14/index.html>
- <https://www.pgadmin.org/>
- [https://www.w3schools.com/sql/sql\\_ref\\_keywords.asp](https://www.w3schools.com/sql/sql_ref_keywords.asp)
- <https://github.com/devrimgunduz/pagila>





Przemysław Wojtysiak

**Trener**

W przypadku pytań lub wątpliwości  
pozostaję do dyspozycji