

# PostgreSQL i DBeaver - Konfigurujemy

## Spis treści

PostgreSQL .....	1
Instalacja .....	1
Uruchomienie i konfiguracja .....	7
Serwer bazy danych .....	10
Tworzenie własnej bazy danych .....	11
Usuwanie bazy danych .....	12
Utworzenie tabeli .....	12
DBeaver .....	14
Instalacja .....	15
Uruchomienie i konfiguracja .....	18

## PostgreSQL

Zanim przejdziemy do dalszych zabaw, upewnijmy się, że mamy odpowiednio skonfigurowane środowisko do pracy. Zaczniemy od instalacji **PostgreSQL**. Część dotycząca instalacji dotyczy Cię tylko jeżeli nie masz zainstalowanej bazy danych **PostgreSQL** u siebie na komputerze.



Obraz 1. PostgreSQL logo. Źródło: <https://icon-library.com/>

**PostgreSQL**: The World's Most Advanced Open Source Relational Database.

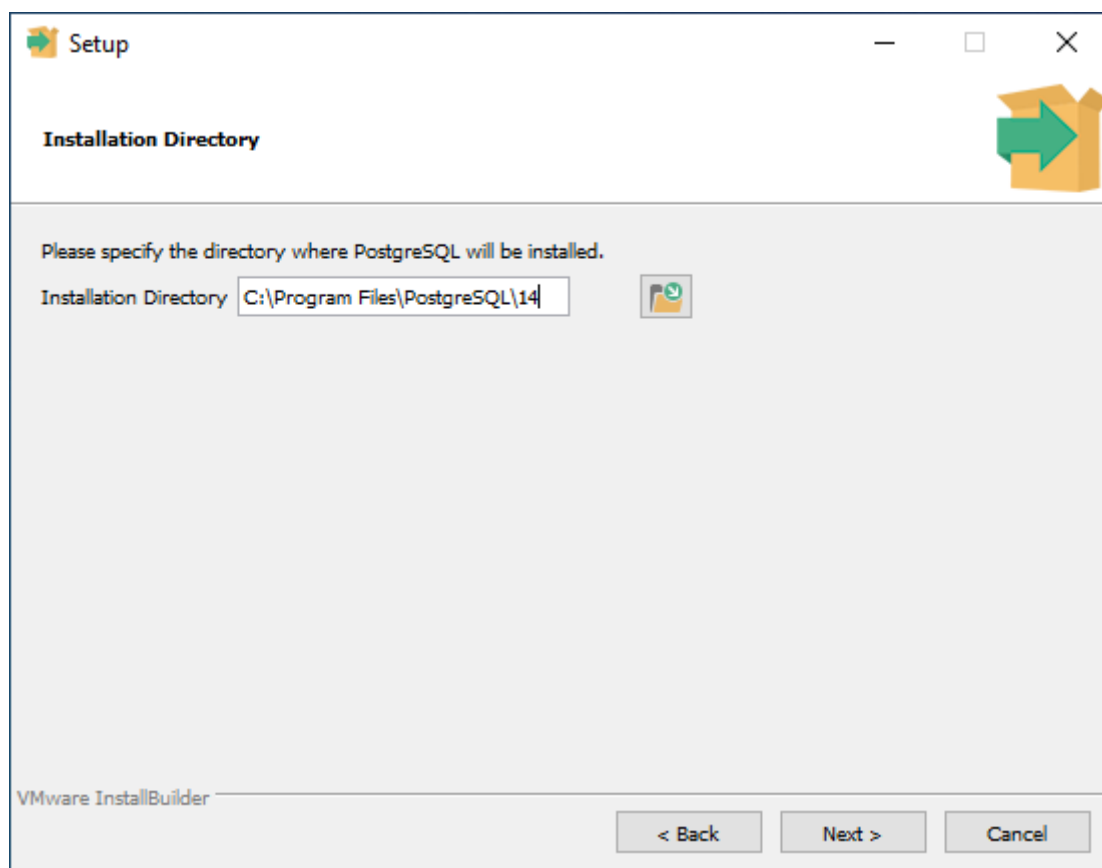
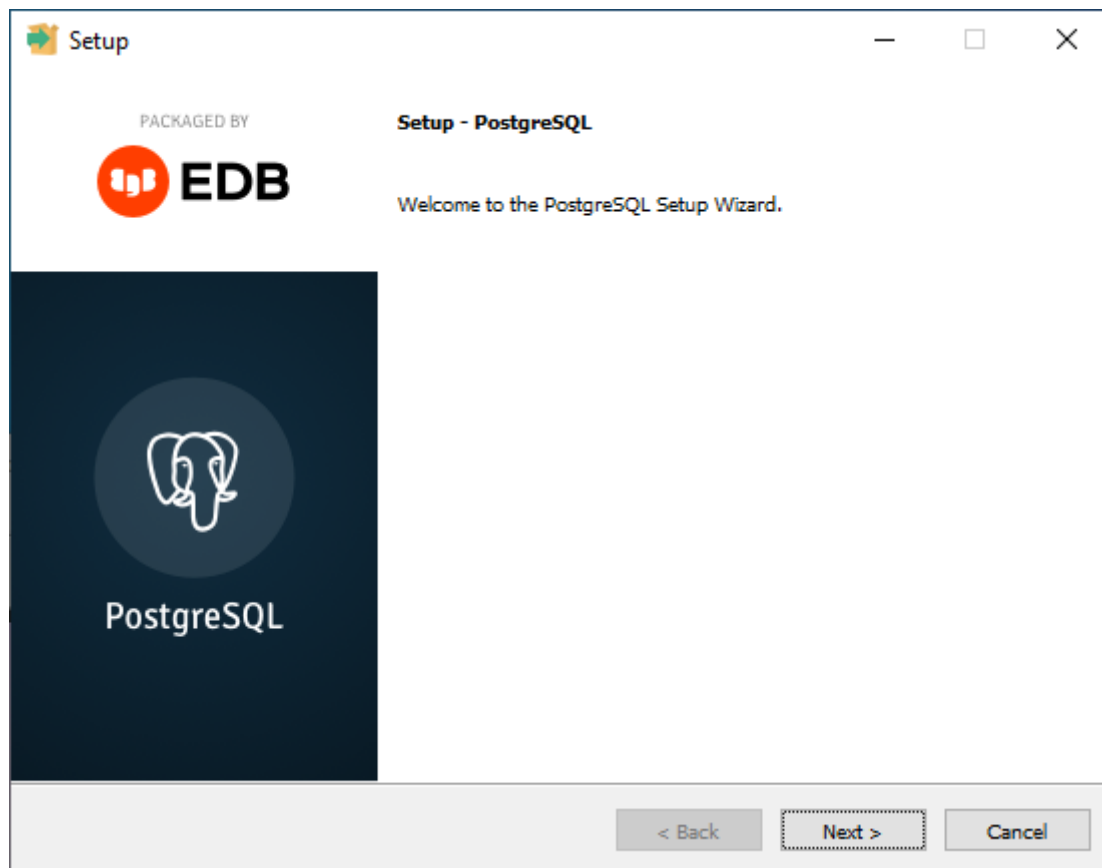
## Instalacja

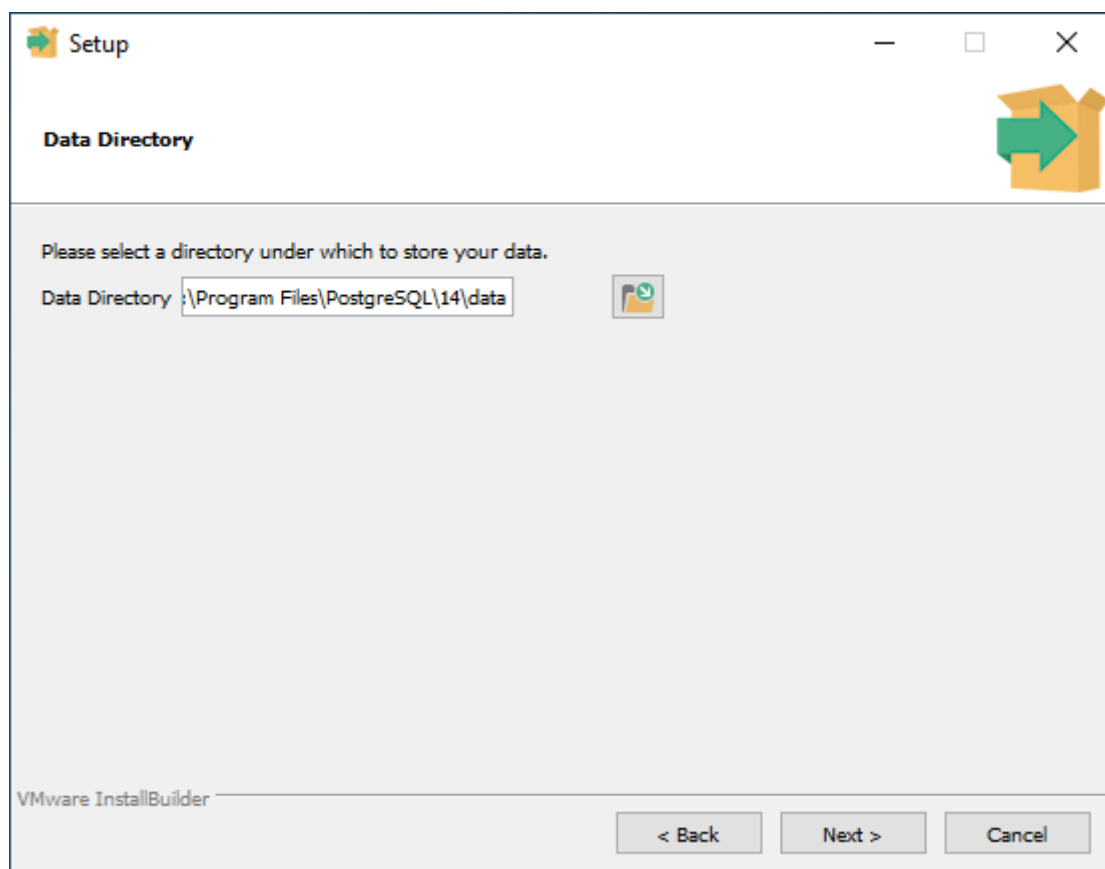
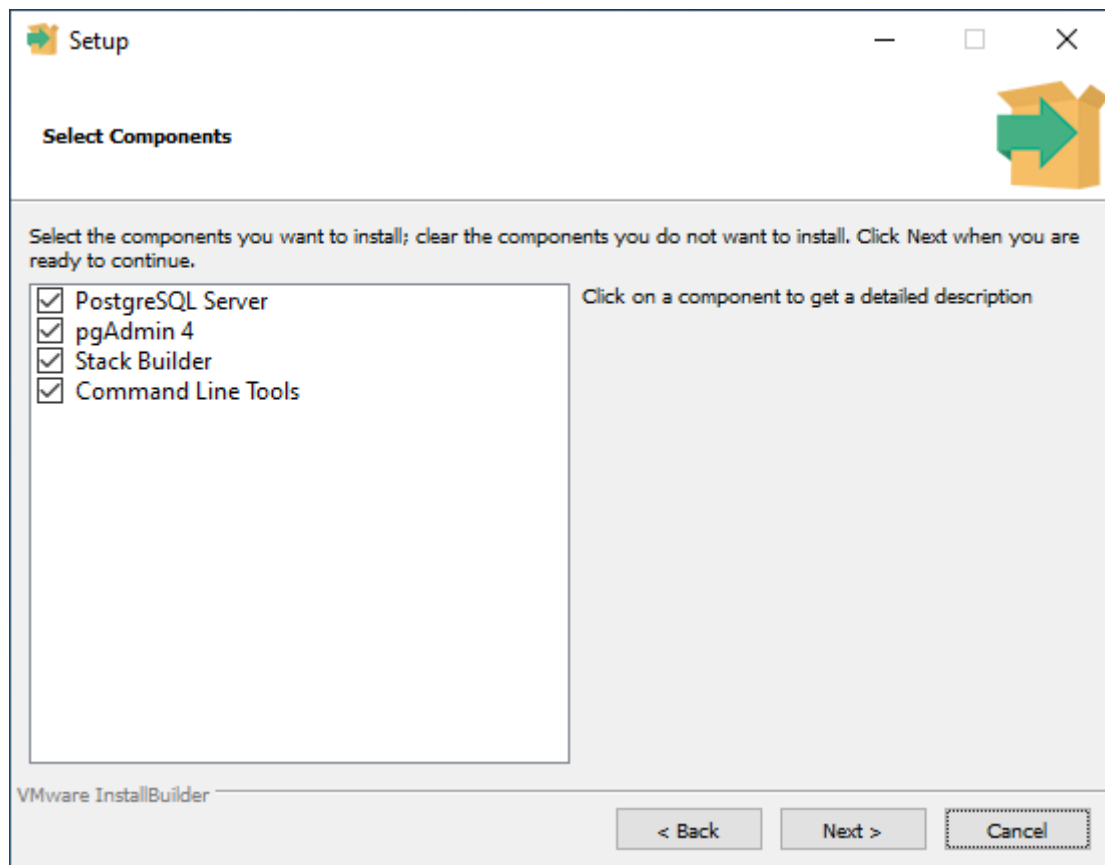
Link do pliku instalacyjnego można pobrać [stąd](#) lub [stąd](#).

W przypadku pracy z systemem Windows, pobieramy plik z rozszerzeniem **.exe** i instalujemy przechodząc przez kolejne kroki. Pamiętajmy, żeby podać hasło, które będziemy pamiętać, żeby potem nie bawić się w resetowanie 😊. **Locale** możemy ustawić na Polskę, wiemy już co to **Locale** jeżeli idziemy

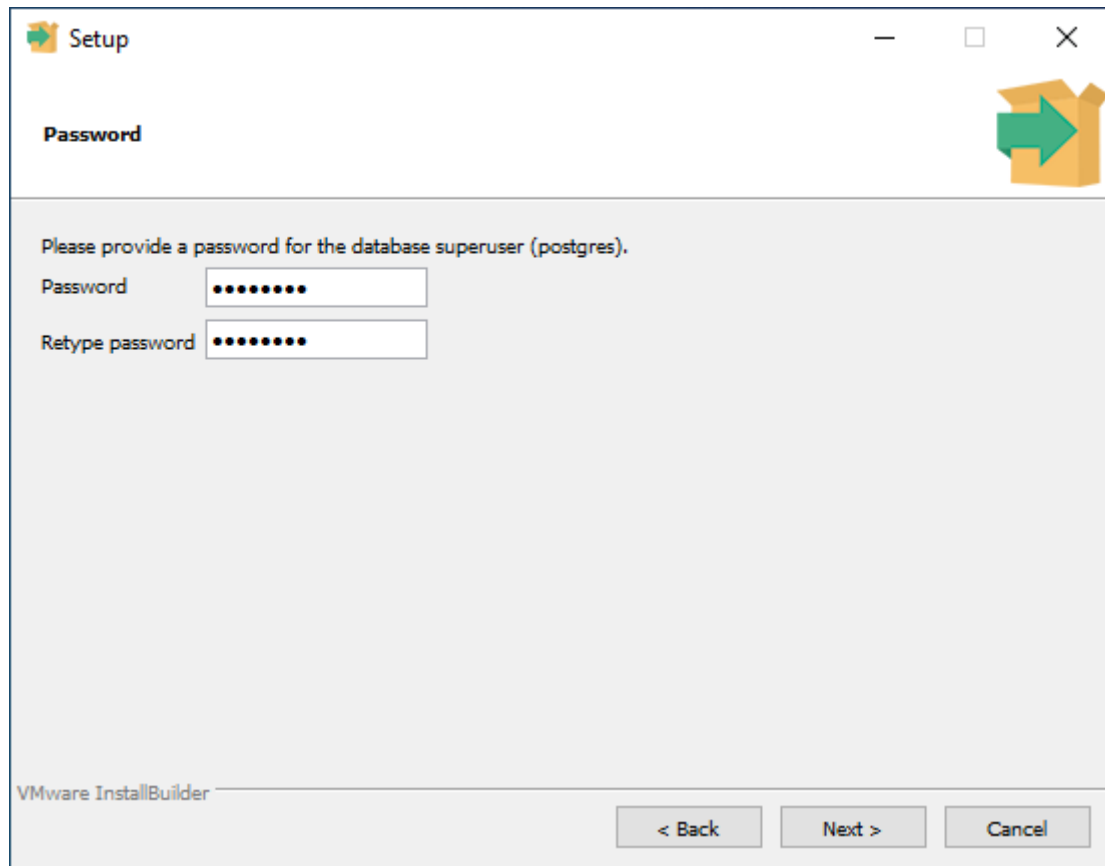
z kolejnością warsztatów. Reszta ustawień może pozostać domyślna, port również zostawiamy 5432. Na końcu może pojawić się pytanie o StackBuilder, które odznaczamy i zrobione.

To samo w formie obrazkowej:





Wpisz hasło np. `password`.



The image shows a 'Setup' window from VMware InstallBuilder. The title bar includes the VMware logo, the word 'Setup', and standard window controls. A large green arrow icon is in the top right corner. The main area is titled 'Password' and contains the instruction 'Please provide a password for the database superuser (postgres)'. Below this are two text input fields: 'Password' and 'Retype password', both containing masked characters (dots). At the bottom, there is a 'VMware InstallBuilder' label and three buttons: '< Back', 'Next >', and 'Cancel'.

**Setup**

**Password**

Please provide a password for the database superuser (postgres).

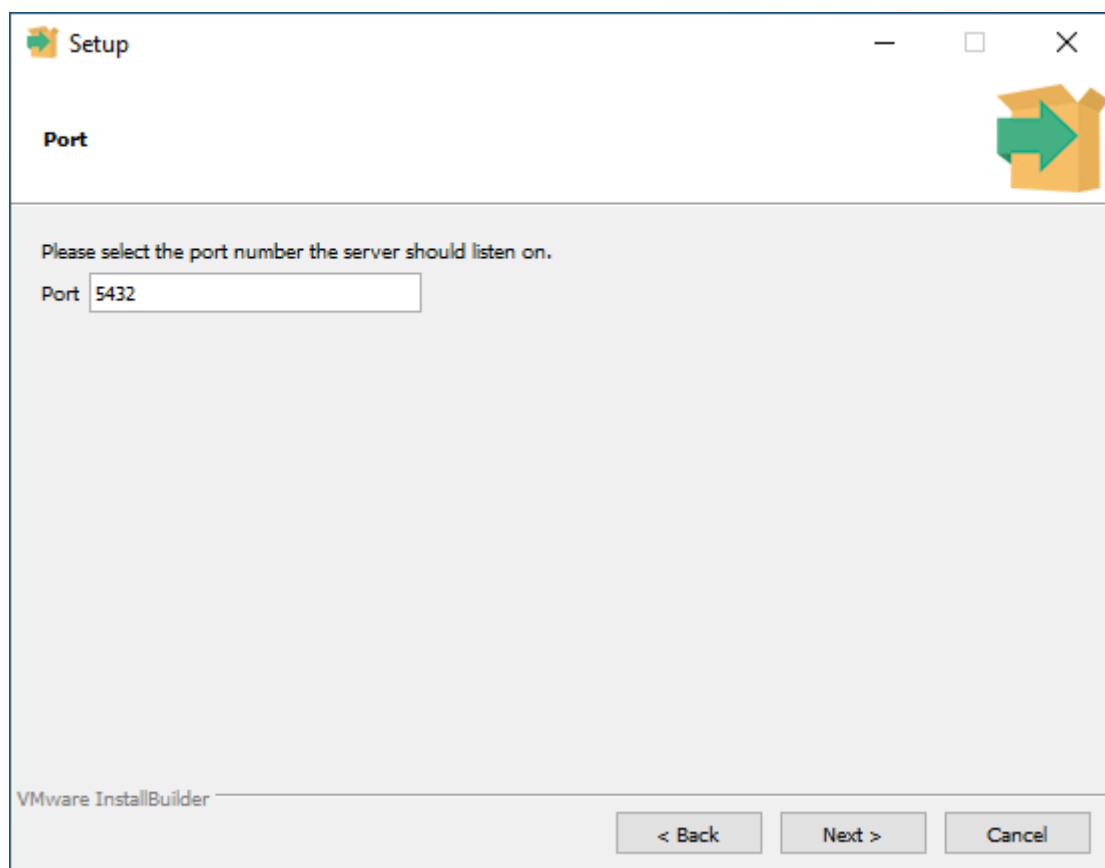
Password

Retype password

VMware InstallBuilder

< Back   Next >   Cancel

Port: 5432



The image shows a 'Setup' window from VMware InstallBuilder. The title bar includes the VMware logo, the word 'Setup', and standard window controls. A large green arrow icon is in the top right corner. The main area is titled 'Port' and contains the instruction 'Please select the port number the server should listen on.'. Below this is a text input field labeled 'Port' containing the value '5432'. At the bottom, there is a 'VMware InstallBuilder' label and three buttons: '< Back', 'Next >', and 'Cancel'.

**Setup**

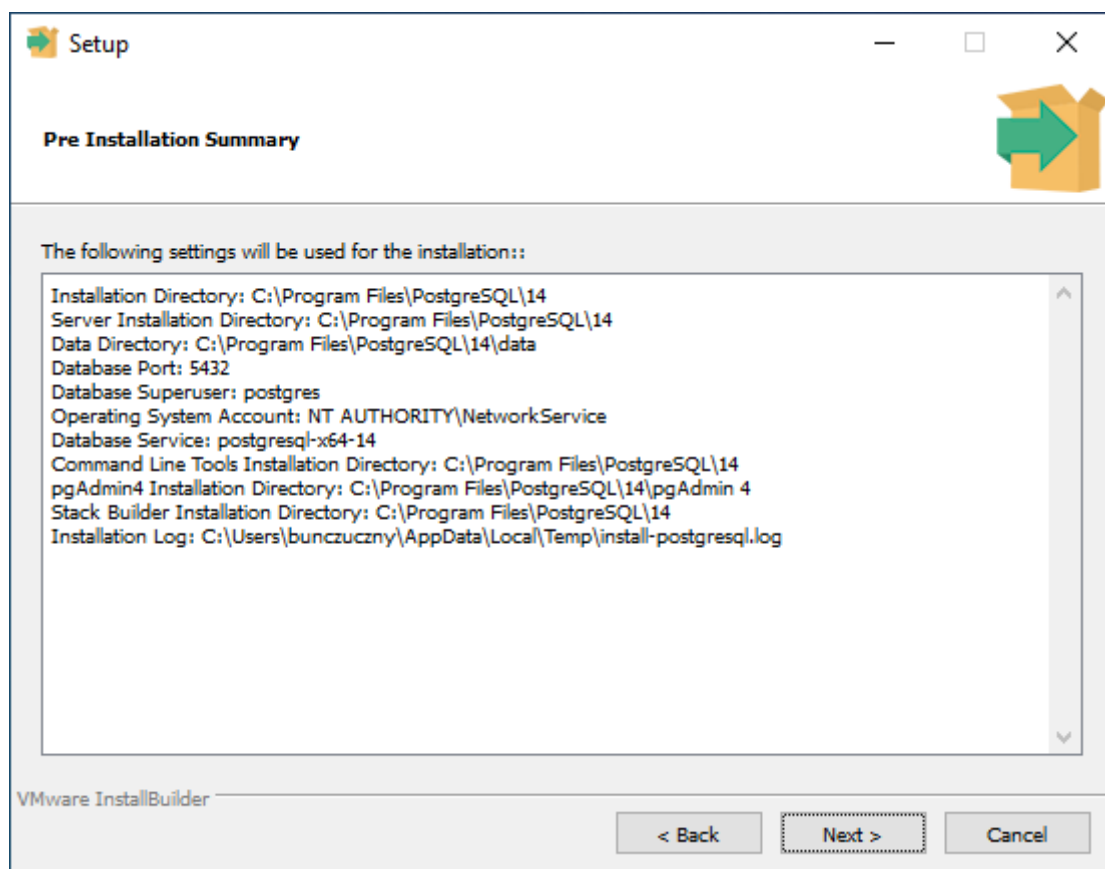
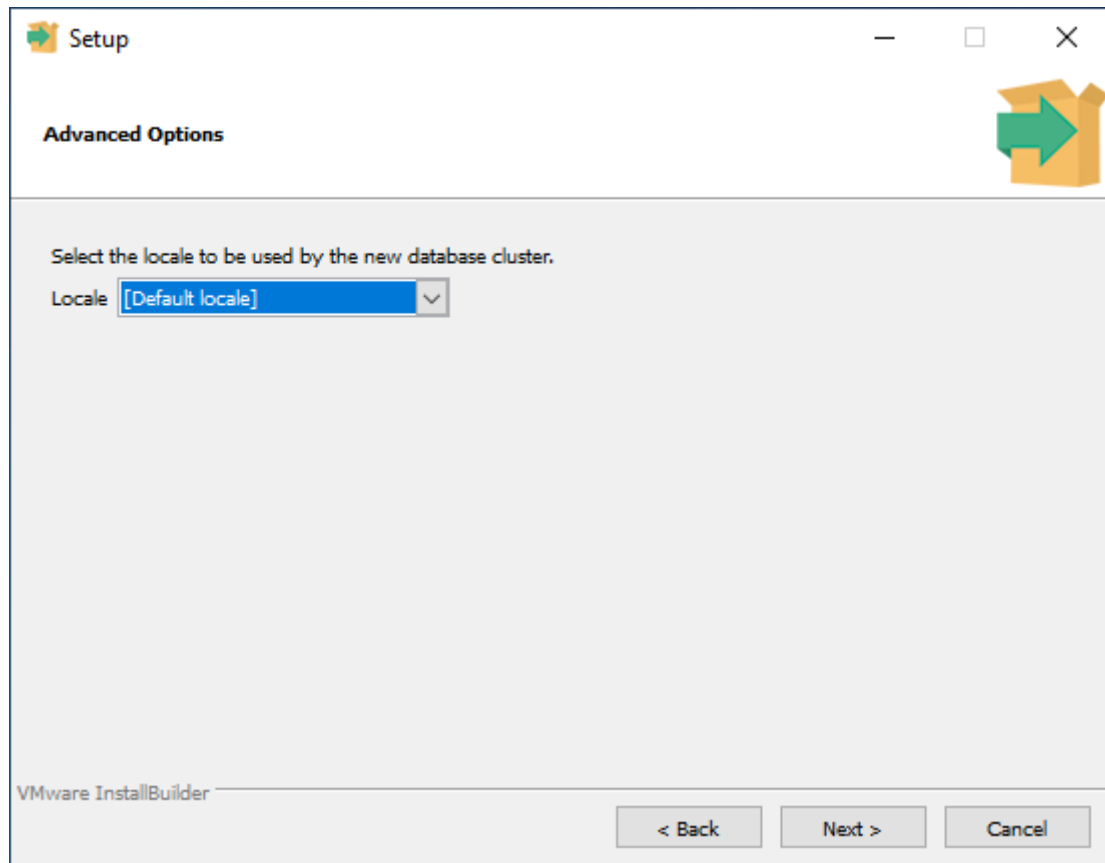
**Port**

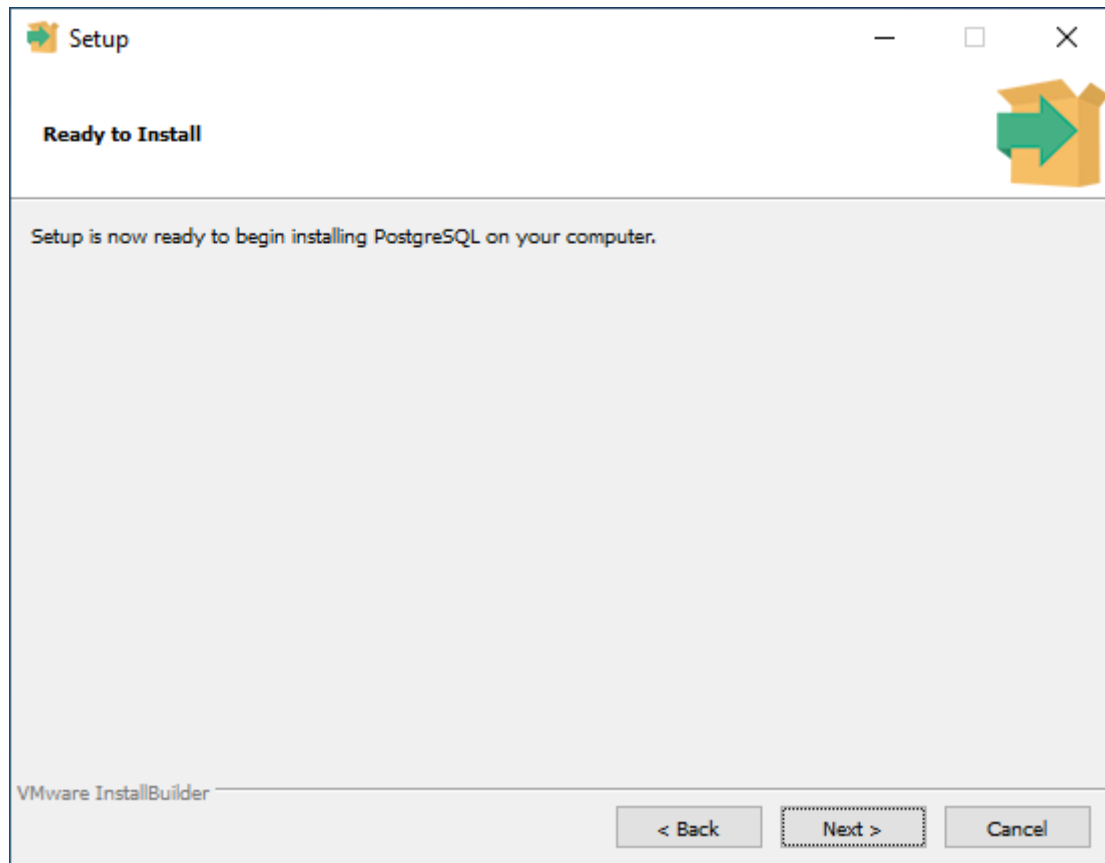
Please select the port number the server should listen on.

Port

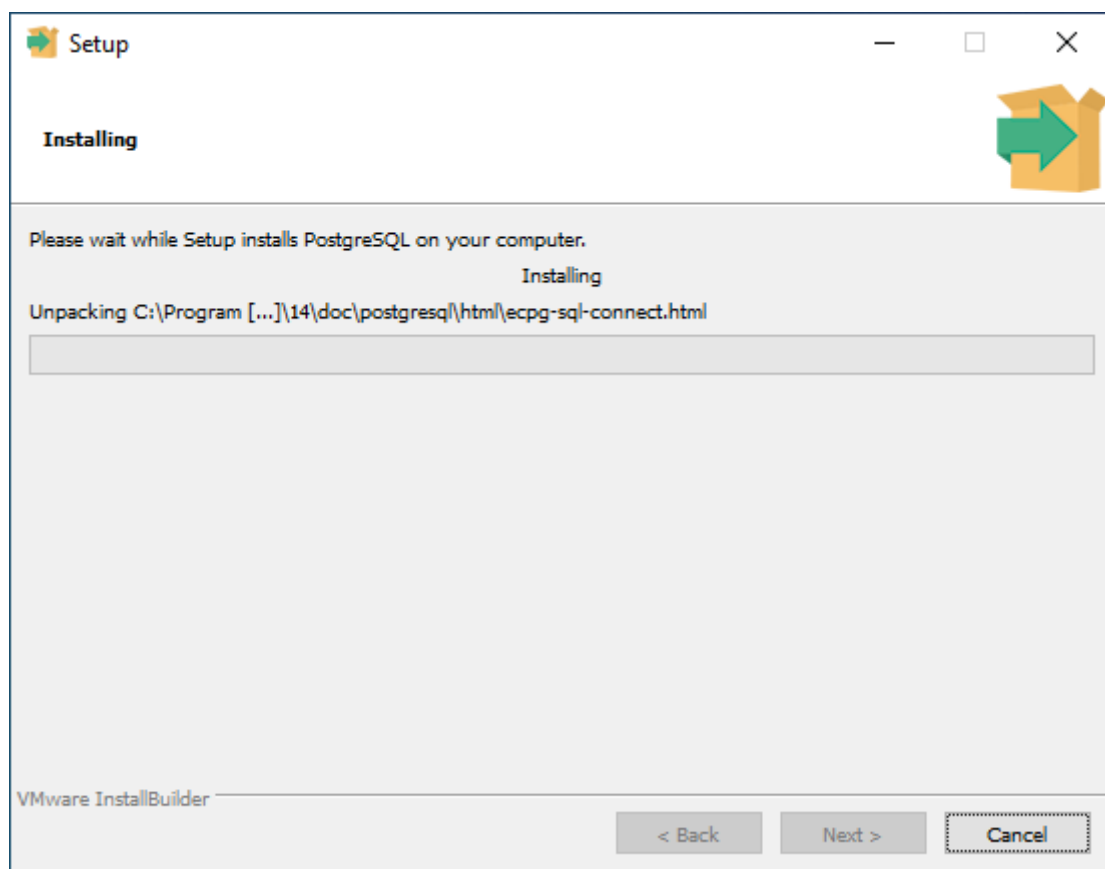
VMware InstallBuilder

< Back   Next >   Cancel

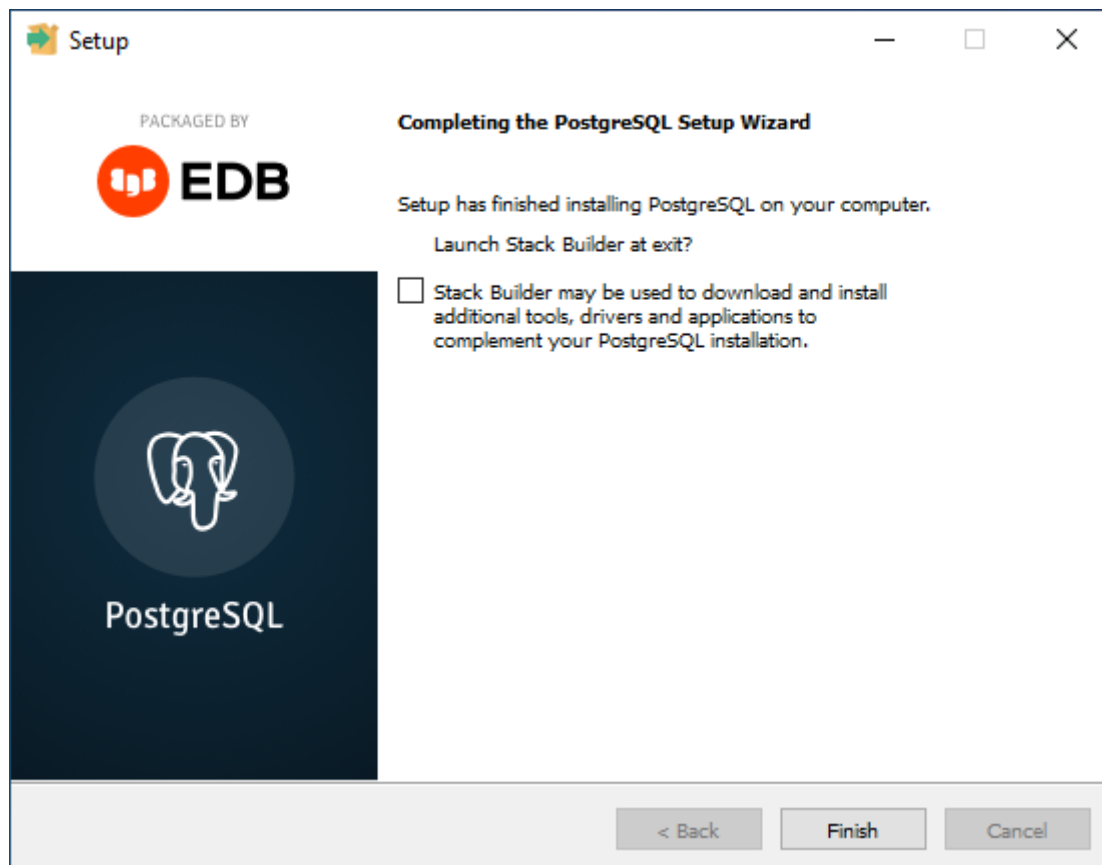




Teraz trzeba poczekać na zakończenie instalacji.



Oznacz checkbox *Stack Builder ...* i wciśnij *Finish*.

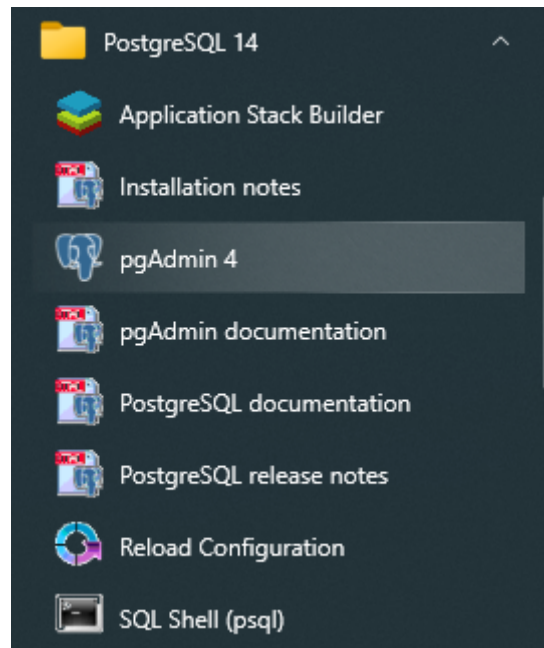


I gotowe 😊.

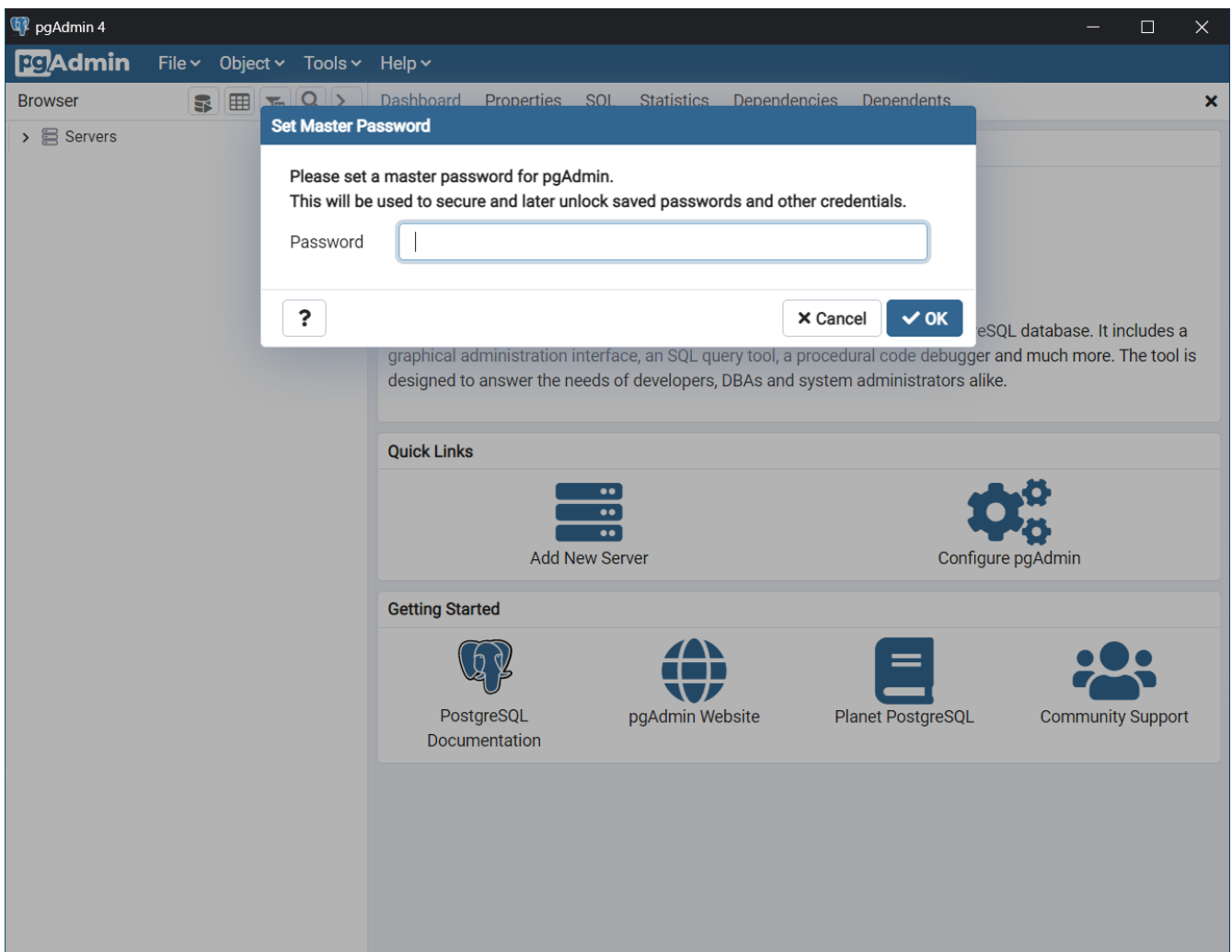
## Uruchomienie i konfiguracja

Teraz aby uruchomić panel do zarządzania należy poszukać aplikacji **pgAdmin**. U mnie jest to **pgAdmin 4**. Jeżeli **pgAdmin** nie chce się poprawnie uruchomić po instalacji można spróbować albo wykonać restart komputera albo uruchomić aplikację jako administrator.

**pgAdmin** może też poprosić o ustawienie **master password**. Z racji, że nasza baza danych nie będzie bazą produkcyjną dla 1mln klientów, możemy ustawić te same hasło co wcześniej przy instalacji. Trzeba też pamiętać, że to co zostanie pokazane w obrębie przedstawionych materiałów ma charakter edukacyjny, w praktyce w systemach produkcyjnych konfiguracja będzie inna i dostosowana do konkretnego przypadku.

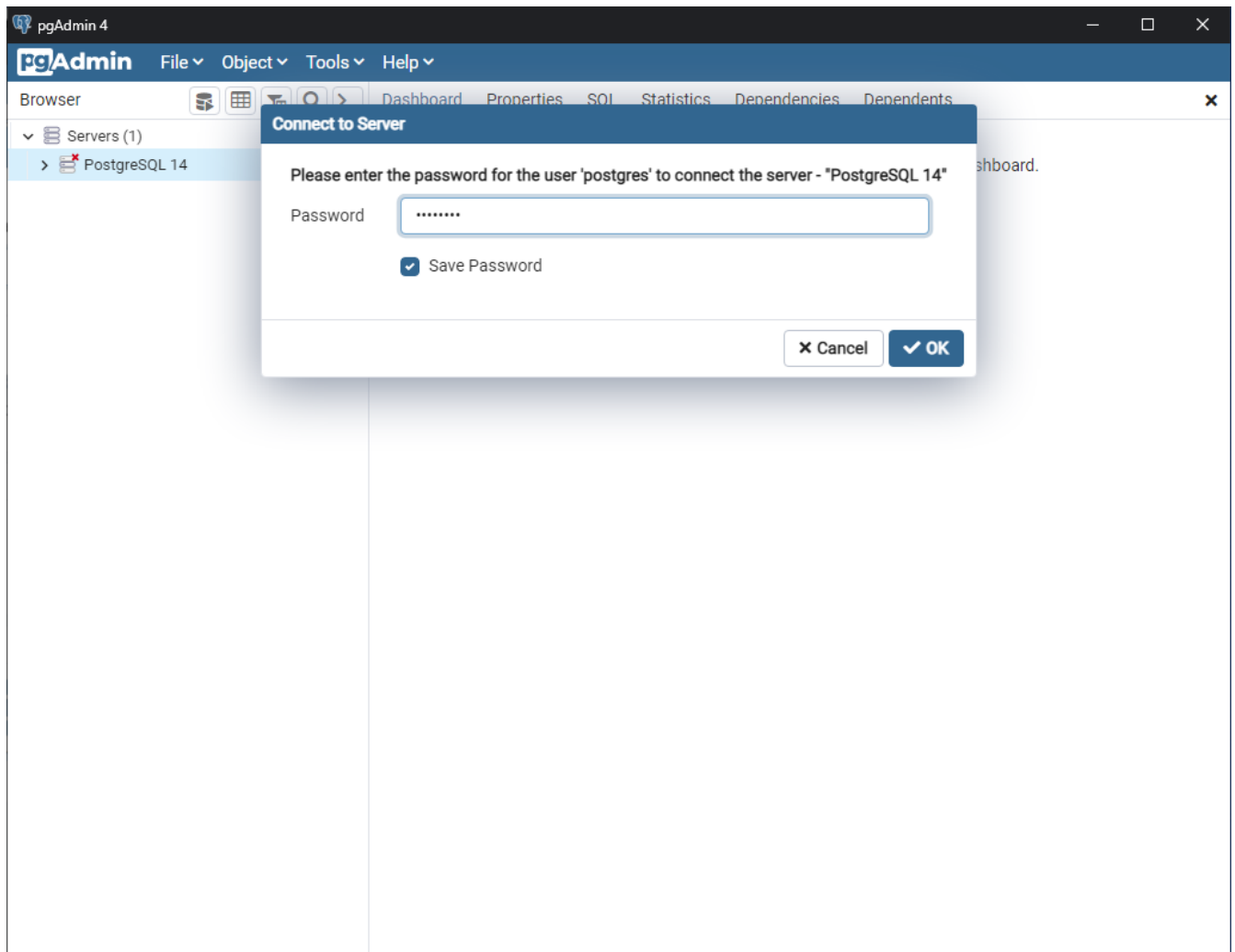


Wpisz wcześniej zdefiniowane hasło, np. *password* i wciśnij *OK*.

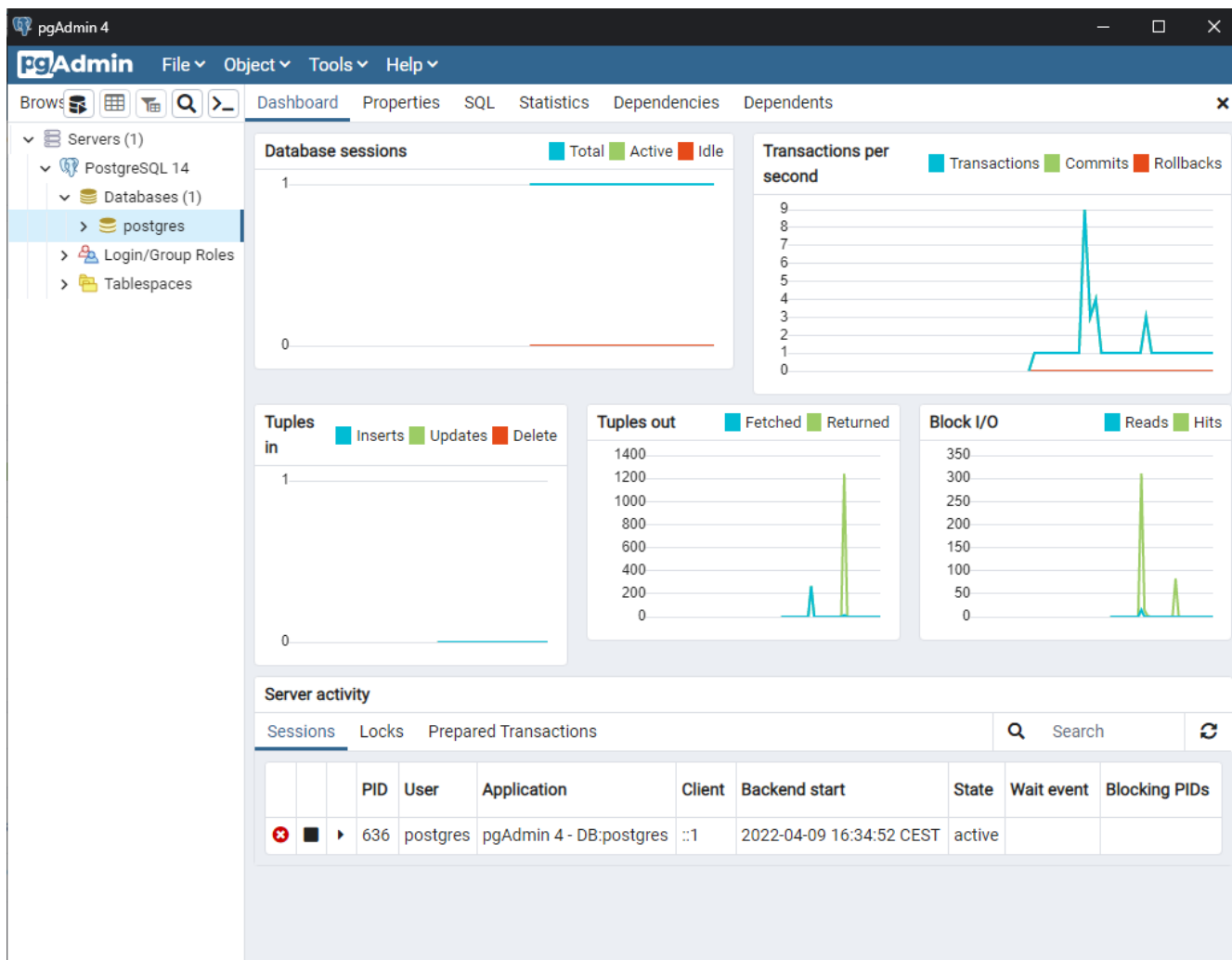


Teraz po lewej stronie rozwiń listę serwerów, wybierz *PostgreSQL 14* i ponownie wpisz hasło.





Brawo! Serwer śmiga.



## Serwer bazy danych

Bardzo przyjemnie ogląda się to co teraz dzieje się na GUI **pgAdmina**, natomiast w praktyce często jest tak, że dużo operacji musi być wykonywane z konsoli. Wynika to z tego, że jeżeli chcemy podłączyć się do serwera bazodanowego, który zapewnia nam bazę danych dla naszej aplikacji, to serwer taki udostępnia przykładowo tylko dostęp z poziomu terminala, a nie daje nam dostępu przez GUI. Z tego powodu, uruchom proszę terminal 😊.

Na Windowsie można wpisać CMD w pasku wyszukiwania dostępnym pod ikoną 'Windows' na klawiaturze, albo z poziomu IntelliJ możemy poszukać Terminal na dole ekranu, lub użyć skrótu **ALT + F12**.

Będziemy korzystać z komendy **psql**, natomiast jeżeli wpiszesz teraz komendę **psql** to dostaniesz informację, że nie rozpoznano takiej komendy. Mamy 2 opcje, albo dodajemy ścieżkę z plikiem **psql.exe** do zmiennej **Path**, albo znajdujemy folder, w którym mamy zainstalowany **PostgreSQL** i znajdujemy tam folder **bin**. Następnie w terminalu wykonujemy komendę (lokalizacja jest przykładowa, Twoja może się różnić):

```
cd C:/Program Files/PostgreSQL/<nr_wersji>/bin
```

Mamy teraz dostęp do komendy **psql**, wykonajmy zatem komendę:

```
psql -U postgres
```

Oznacza ona próbę połączenia się z serwerem baz danych PostgreSQL przy wykorzystaniu użytkownika **postgres**. Ustawialiśmy hasło temu użytkownikowi na etapie instalacji PostgreSQL. Zostaniemy poproszeni o hasło podane podczas instalacji. Na ekranie pojawi się:

```
postgres=#
```

Zwróć uwagę, że znika ścieżka systemu Windows z której próbowaliśmy się dostać. Oznacza to, że udało nam się połączyć do serwera bazy danych. Aby wydostać się z serwera wystarczy napisać **quit**.

Możemy w tym momencie wylistować utworzonych użytkowników za pomocą komendy:

```
\du
```

I zobaczymy, że mamy dostępnego tylko użytkownika **postgres**. Na ten moment nam to wystarczy.

Możemy teraz wylistować dostępne bazy danych za pomocą komendy:

```
\l
```

Aby pobrać trochę więcej informacji, możemy wpisać:

```
\l+
```

Widzimy teraz domyślnie utworzoną bazę danych **postgres**, ale zaraz będziemy tworzyć swoją. Aby zobaczyć dostępne bazy danych, możemy równie dobrze napisać:

```
SELECT datname FROM pg_database;
```

## Tworzenie własnej bazy danych

Wystarczy, że wpiszemy komendę:

```
CREATE DATABASE zajavka;
```

Możemy sprawdzić, czy udało nam się utworzyć bazę danych stosując ponownie komendę:

```
\l+
```

Taka baza danych będzie też teraz widoczna w **pgAdminie**. Na ikonie **servers** klikamy prawym przyciskiem myszki, wybieramy **Refresh** i po rozwinięciu **Databases** zobaczymy poprzednią bazę

danych oraz bazę danych **zajavka**.

Jeżeli na którymś etapie będziemy szukać odpowiedzi na nurtujące nas kwestie, polecam **dokumentację**. Możemy tutaj znaleźć opis komend, które stosujemy, przykładowo komenda **CREATE DATABASE**.

Wykorzystaliśmy najprostszy wariant komendy, gdzie bardzo duża ilość ustawień została wybrana domyślnie, ale na ten moment nie przejmujemy się tym. Prawdopodobnie zaczniesz się tego uczyć dopiero jak będzie Ci to potrzebne w praktyce w pracy. Nikt też nie uczy się tych wszystkich opcji na pamięć (przynajmniej ja nie znam), od tego jest dokumentacja, także uspokajam 😊.

## Usuwanie bazy danych

Taką bazę danych możemy też usunąć stosując komendę **DROP DATABASE**:

```
DROP DATABASE zajavka;
```

Jeżeli otrzymasz informację zwrotną, mówiącą, że jeszcze ktoś korzysta z tej bazy danych, sprawdź czy nie robi tego przypadkiem **pgAdmin**.

## Utworzenie tabeli

W celu kontroli poprawnego działania, stwórz tabelę **person**:

```
CREATE TABLE person
(
    person_id SERIAL NOT NULL,
    age INT,
    name VARCHAR(255),
    PRIMARY KEY (person_id)
);
```

Możesz to zrobić z poziomu **pgAdmin** lub z poziomu konsoli. Tabela powinna być utworzona. Jak to sprawdzić z poziomu konsoli?

Wpisz komendę **\d person**, która wyświetla definicję wskazanej tabeli.

```
postgres=# \d person
               Table "public.person"
  Column   |      Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
 person_id | integer         |           | not null |
    age    | integer         |           |          |
    name    | character varying(255) |           |          |
Indexes:
    "person_pkey" PRIMARY KEY, btree (person_id)

postgres=#
```

Równie dobrze możesz wykonać **SELECT**:

```
SELECT * FROM person;
```

```
postgres=# select * from person;
 person_id | age | name
-----+-----+-----
(0 rows)

postgres=#
```

W **pgAdmin 4** też można to zobaczyć, np. na liście **Tables** w bazie **postgres**.

The screenshot shows the pgAdmin 4 interface. On the left, the 'Browser' pane displays the database structure: Servers (1) > PostgreSQL 14 > Databases (1) > postgres > Schemas (1) > public > Tables (1) > person. The 'person' table is selected, showing its columns: person\_id, age, and name.

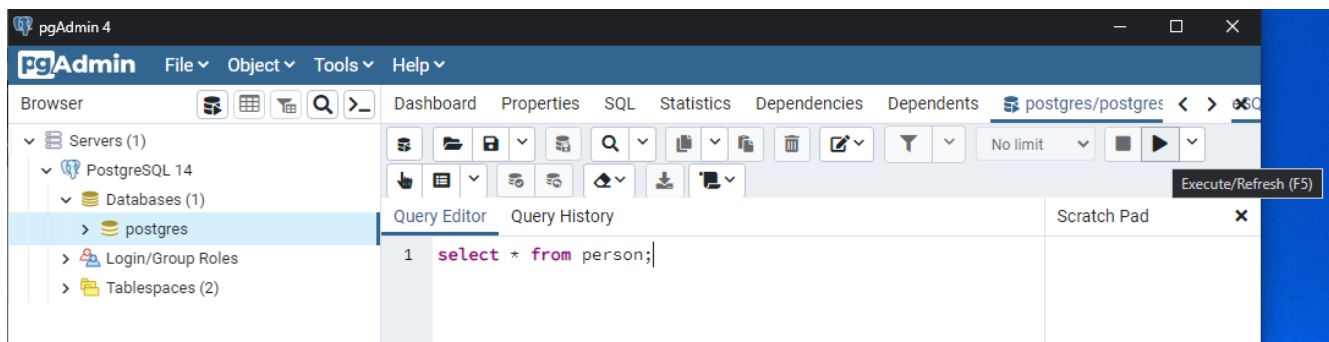
The main panel displays several performance charts:

- Database sessions**: A line chart showing Total (blue), Active (green), and Idle (red) sessions over time. The y-axis ranges from 0 to 2.
- Transactions per second**: A line chart showing Transactions (blue), Commits (green), and Rollbacks (red) over time. The y-axis ranges from 0 to 3.
- Tuples in**: A line chart showing Inserts (blue), Updates (green), and Deletes (red) over time. The y-axis ranges from 0 to 1.
- Tuples out**: A line chart showing Fetched (blue) and Returned (green) tuples over time. The y-axis ranges from 0 to 1400.
- Block I/O**: A line chart showing Reads (blue) and Hits (green) over time. The y-axis ranges from 0 to 90.

At the bottom, the 'Server activity' section shows a table of active sessions:

	PID	User	Application	Client	Backend start	State	Wait event
✖	636	postgres	pgAdmin 4 - DB:postgres	::1	2022-04-09 16:34:52 CEST	active	
✖	4156	postgres	psql	::1	2022-04-09 16:41:46 CEST	idle	Client: ClientR

Albo za pomocą *Query Tool*, który jest dostępny w **pgAdmin 4**



## DBeaver

Chcę tutaj przedstawić jedno narzędzie, które może być przydatne, jeżeli nie chcesz korzystać z **pgAdmina**. Generalnie narzędzi graficznych, które pozwalają na podłączenie się do bazy danych, żeby móc z nią pracować "graficznie" jest dużo, przykładowo:

- pgAdmin,
- DBeaver,
- IntelliJ Database tool window - [link](#). IntelliJ pozwala na pracę z bazami danych w wersji Ultimate, my korzystamy z Community.
- i więcej...

**DBeaver** możemy pobrać [stąd](#):

**DBeaver**: Free multi-platform database tool for developers, database administrators, analysts and all people who need to work with databases. Supports all popular databases: MySQL, **PostgreSQL**, SQLite, Oracle, DB2, SQL Server, Sybase, MS Access, Teradata, Firebird, Apache Hive, Phoenix, Presto, etc.



# DBeaver Community

Free Universal Database Tool

Star 27,213
Follow @dbeaver\_news

Go

[Home](#)
[About](#)
[Download](#)
[Sources](#)
[Documentation](#)
[News](#)
[Support](#)
[DBeaver PRO](#)
[CloudBeaver](#)

## Universal Database Tool

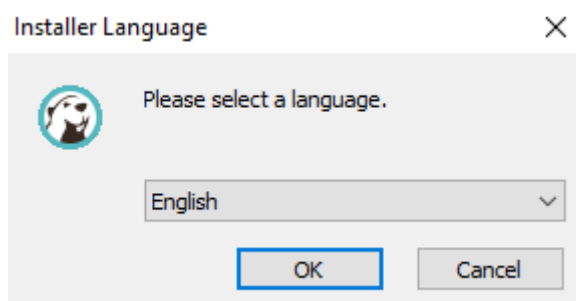
Free multi-platform database tool for developers, database administrators, analysts and all people who need to work with databases. Supports all popular databases: MySQL, PostgreSQL, SQLite, Oracle, DB2, SQL Server, Sybase, MS Access, Teradata, Firebird, Apache Hive, Phoenix, Presto, etc.

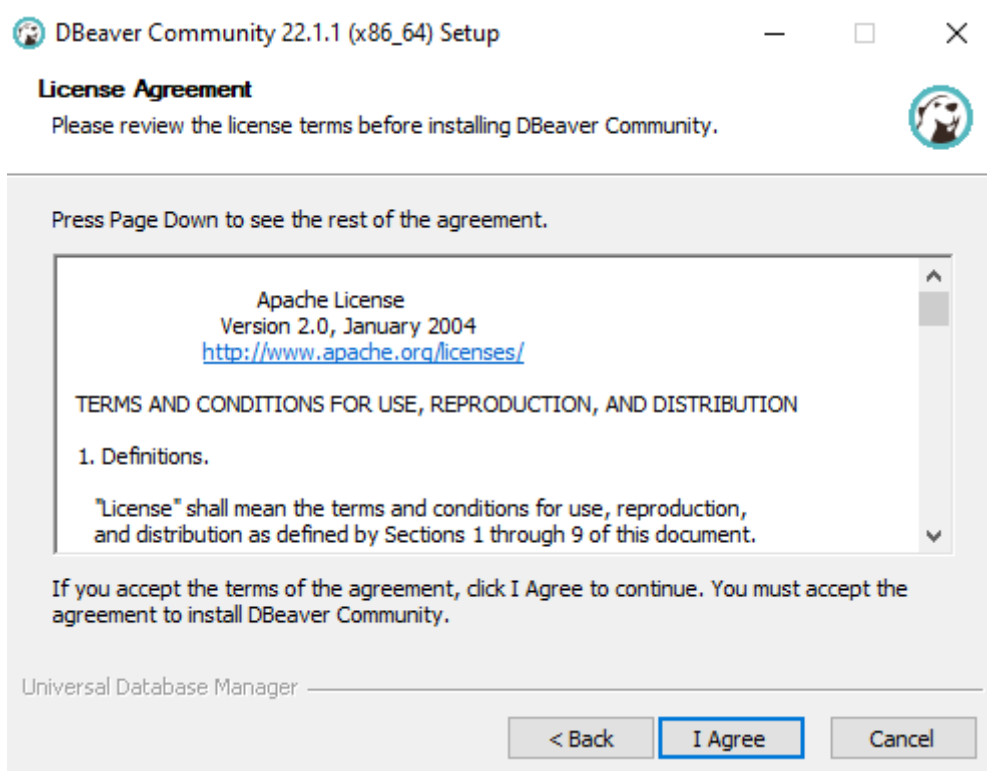
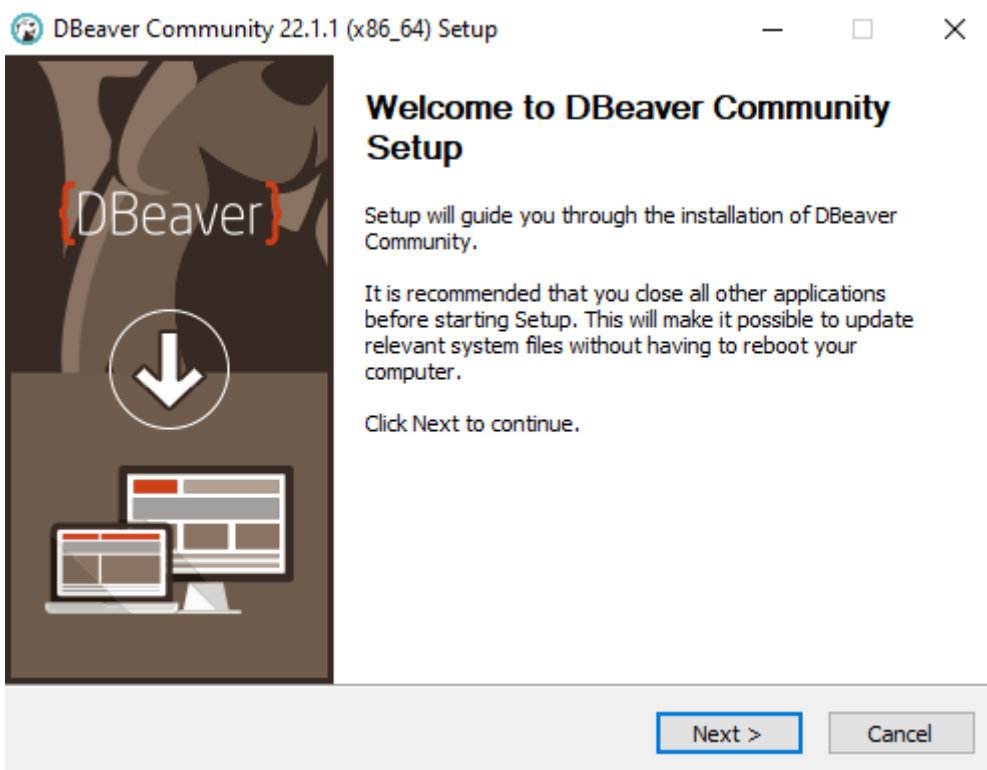
Download



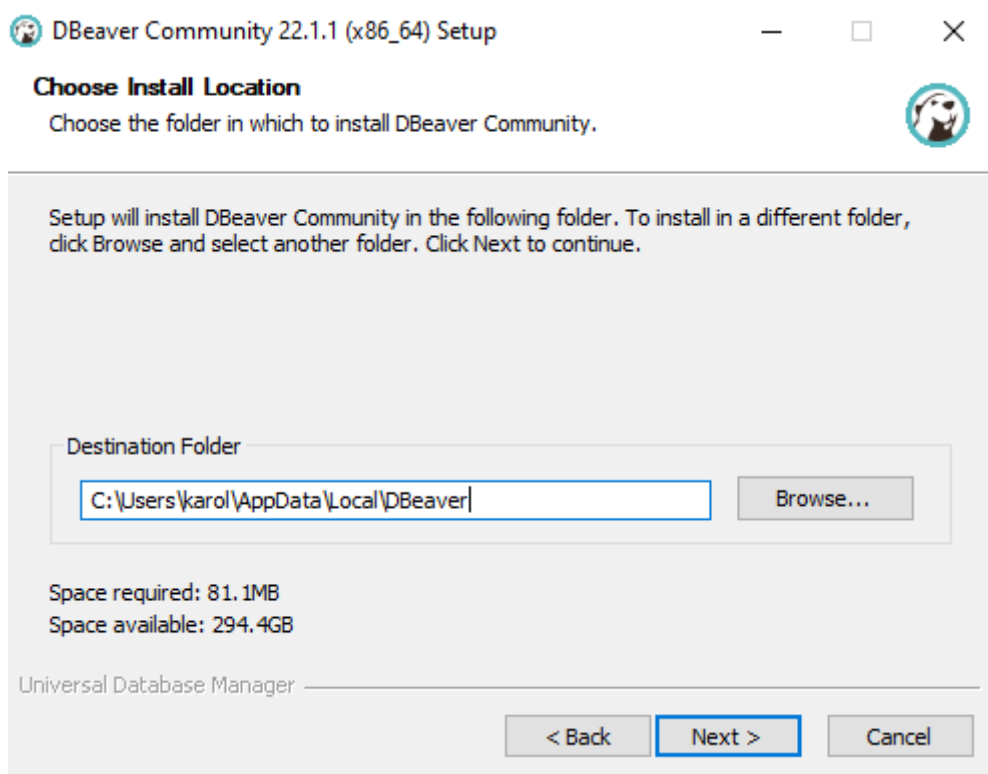
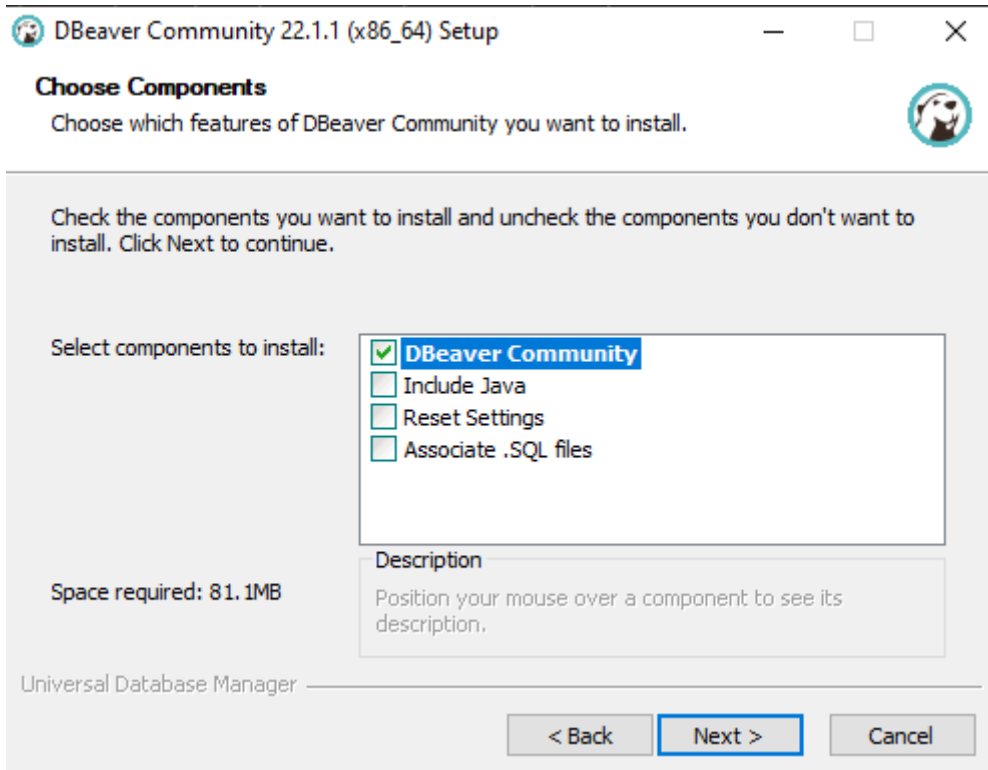
## Instalacja

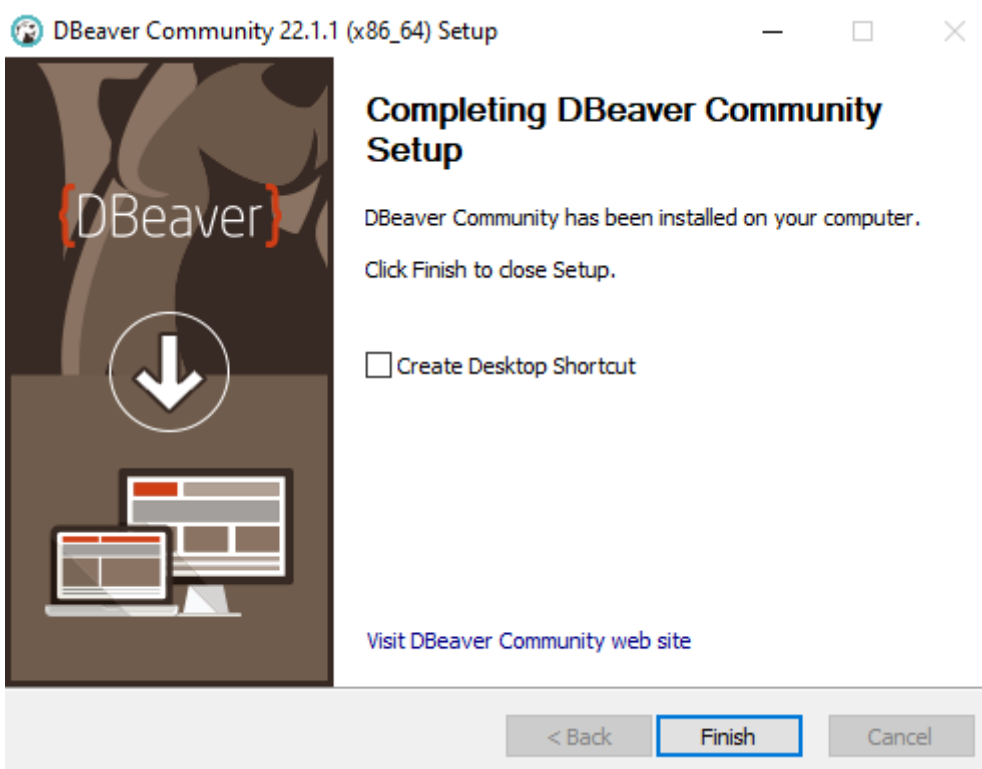
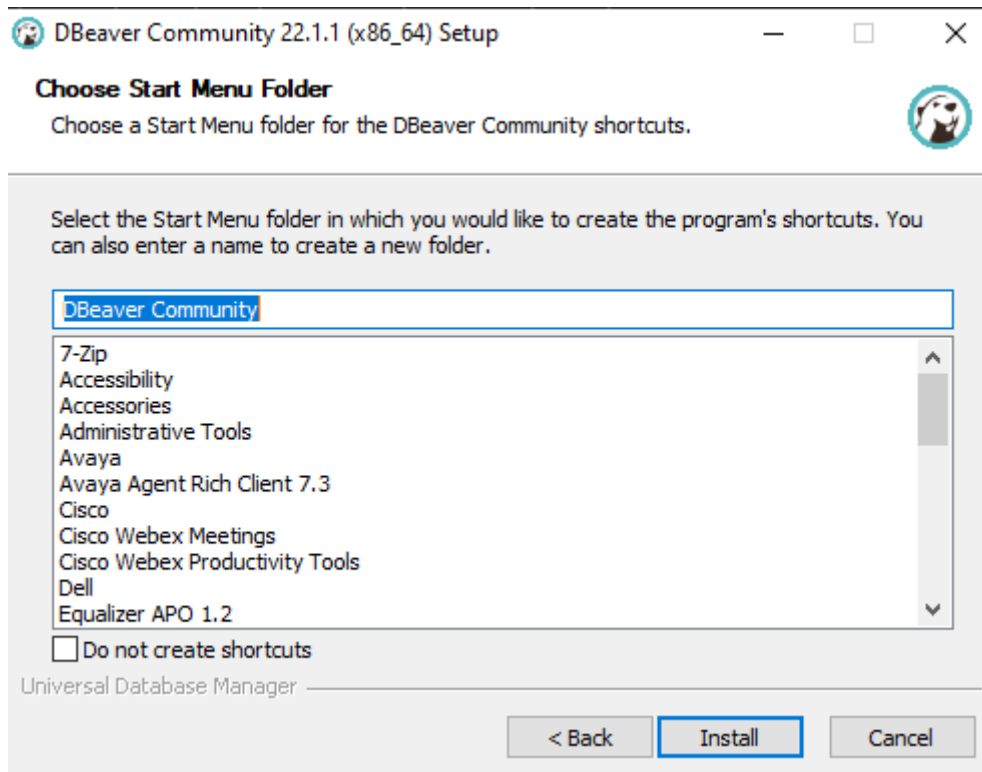
Instalacja wygląda standardowo, poniżej przykład dla Windows.











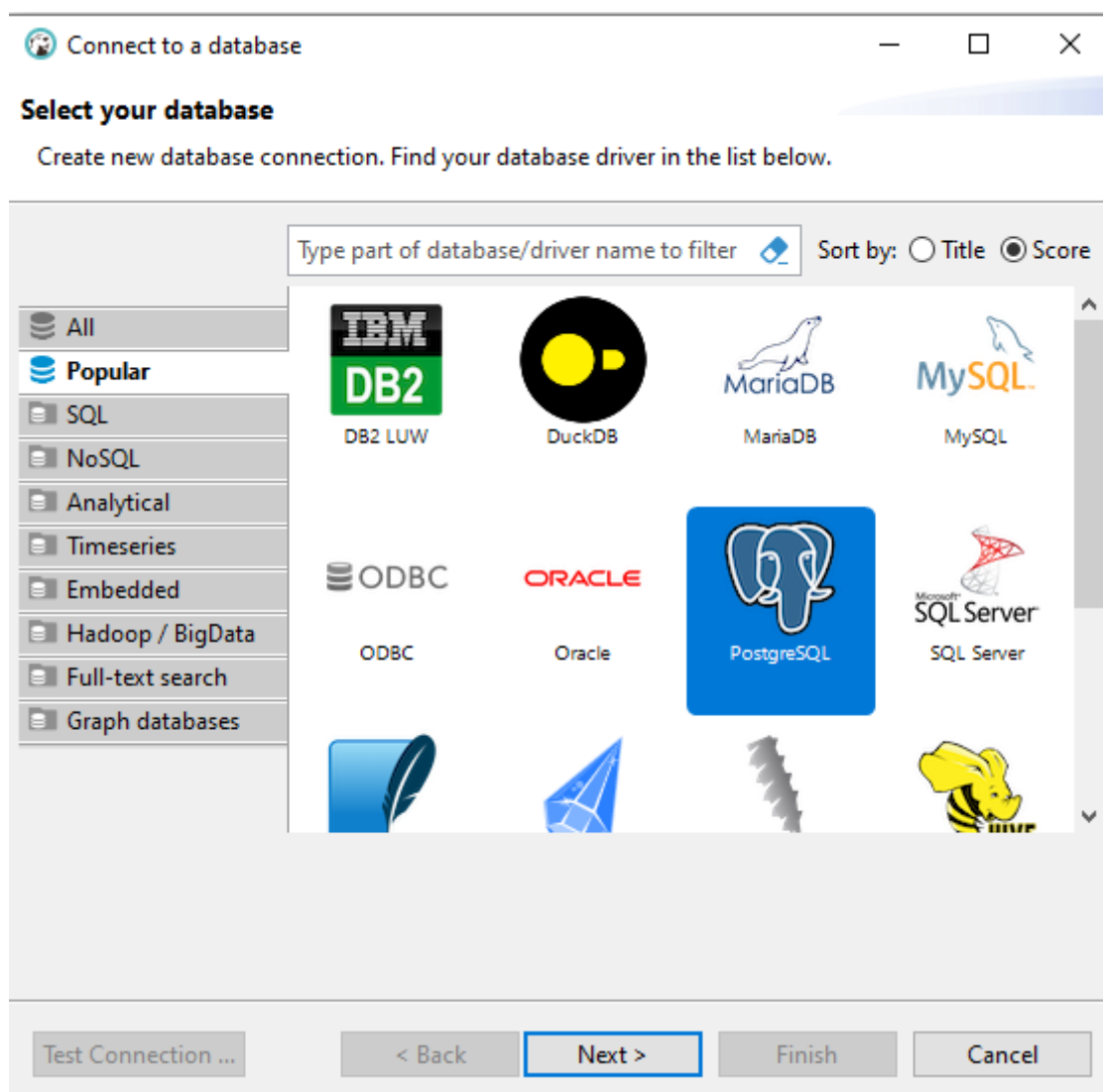
Gdy instalacja zostanie zakończona, możemy uruchomić **DBEaver**.

## Uruchomienie i konfiguracja

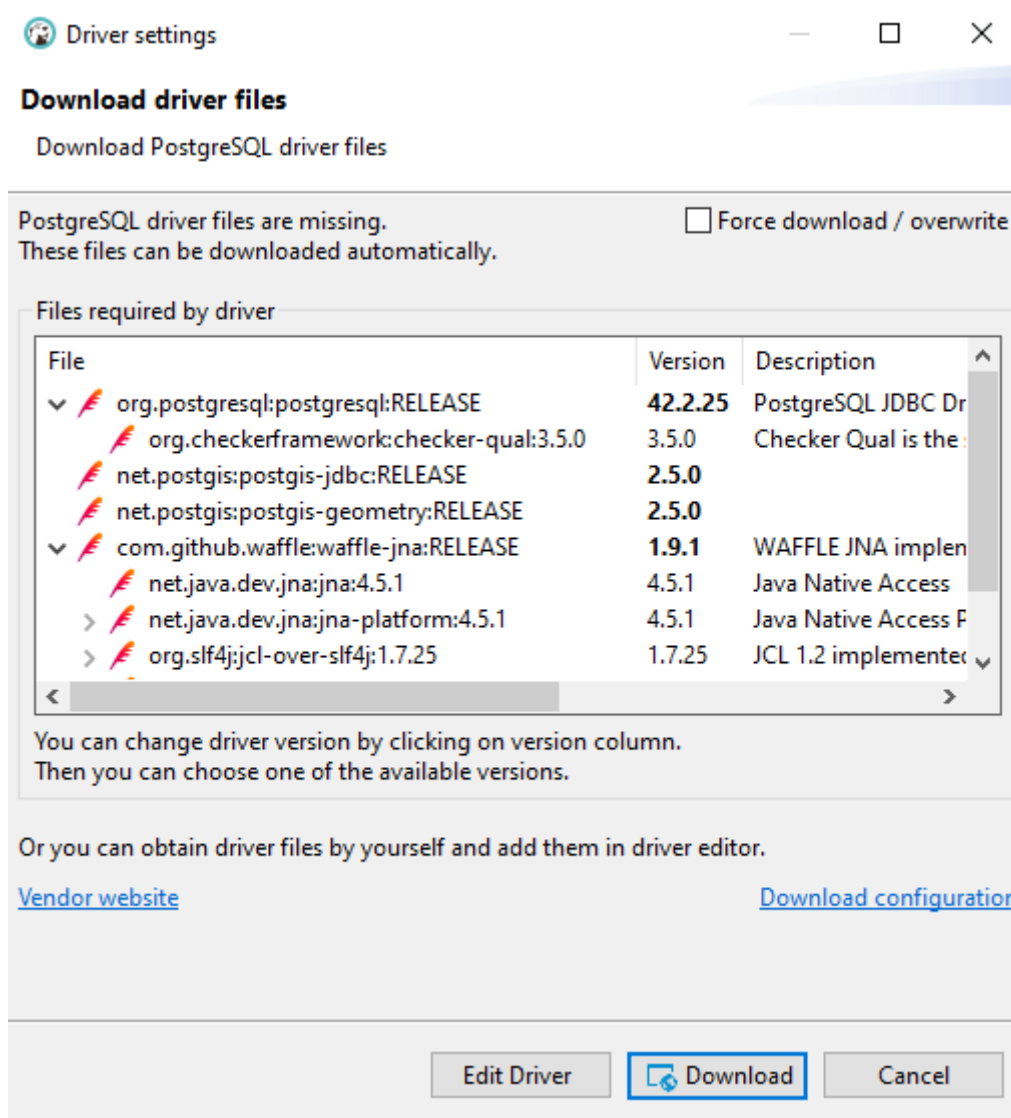
Bardzo lubię te grafiki pokazywane przy uruchamianiu kolejnych narzędzi 😊.



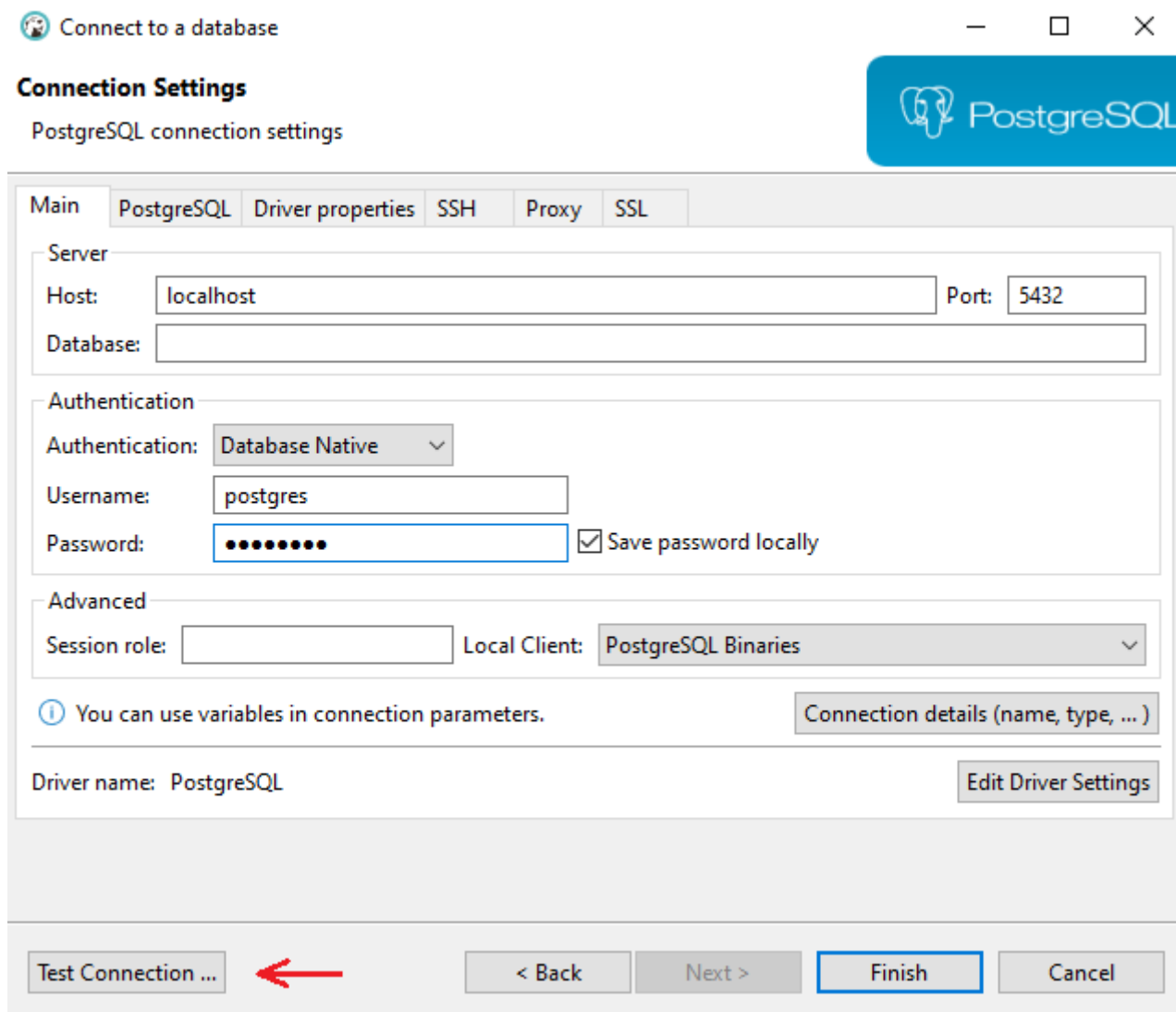
Zaznaczamy, że chcemy dodać połączenie do PostgreSQL:



Możemy zostać poinformowani, że musimy pobrać dodatkowe sterowniki:



Gdy przejdziemy dalej, będziemy mogli dodać konfigurację:



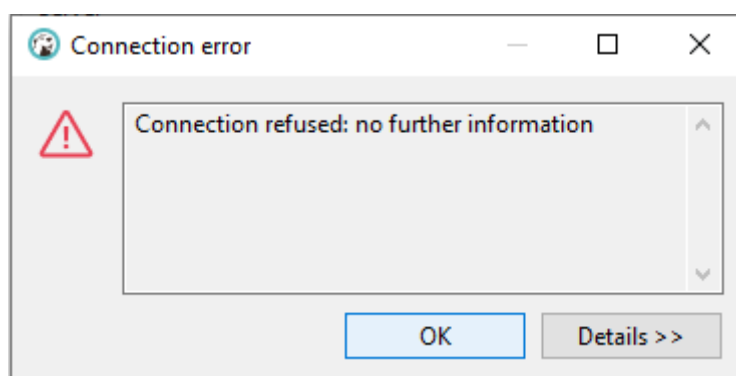
The screenshot shows the 'Connect to a database' dialog box with the 'PostgreSQL' tab selected. The 'Main' sub-tab is active, showing the following settings:

- Server:** Host: localhost, Port: 5432, Database: (empty)
- Authentication:** Authentication: Database Native (dropdown), Username: postgres, Password: (masked with dots), ☒ Save password locally
- Advanced:** Session role: (empty), Local Client: PostgreSQL Binaries (dropdown)

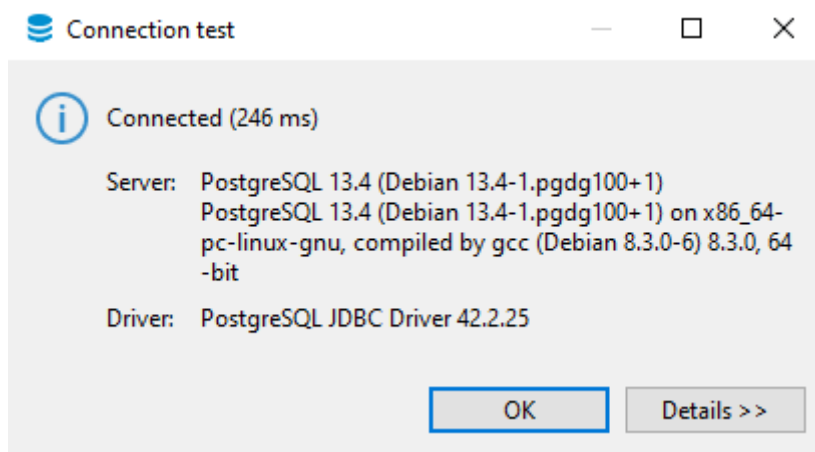
Below the settings, there is a note: 'You can use variables in connection parameters.' and a button 'Connection details (name, type, ...)'. At the bottom, there is a 'Driver name: PostgreSQL' label and an 'Edit Driver Settings' button.

At the very bottom of the dialog, there are five buttons: 'Test Connection ...', '< Back', 'Next >', 'Finish', and 'Cancel'. A red arrow points to the 'Test Connection ...' button.

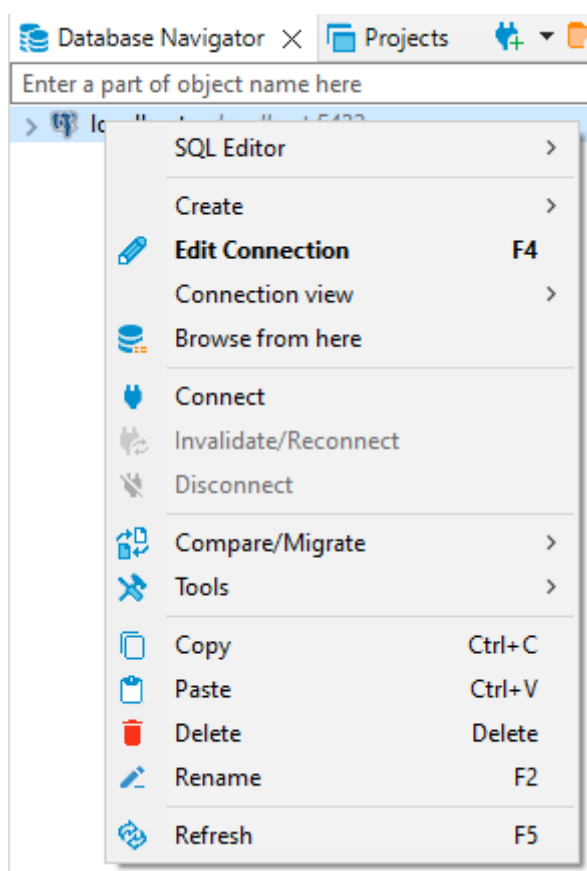
Żeby upewnić się, że wszystko dodaliśmy poprawnie, możemy kliknąć "Test Connection ...". Jeżeli połączenie jest skonfigurowane nieprawidłowo, zobaczymy ekran podobny do tego:



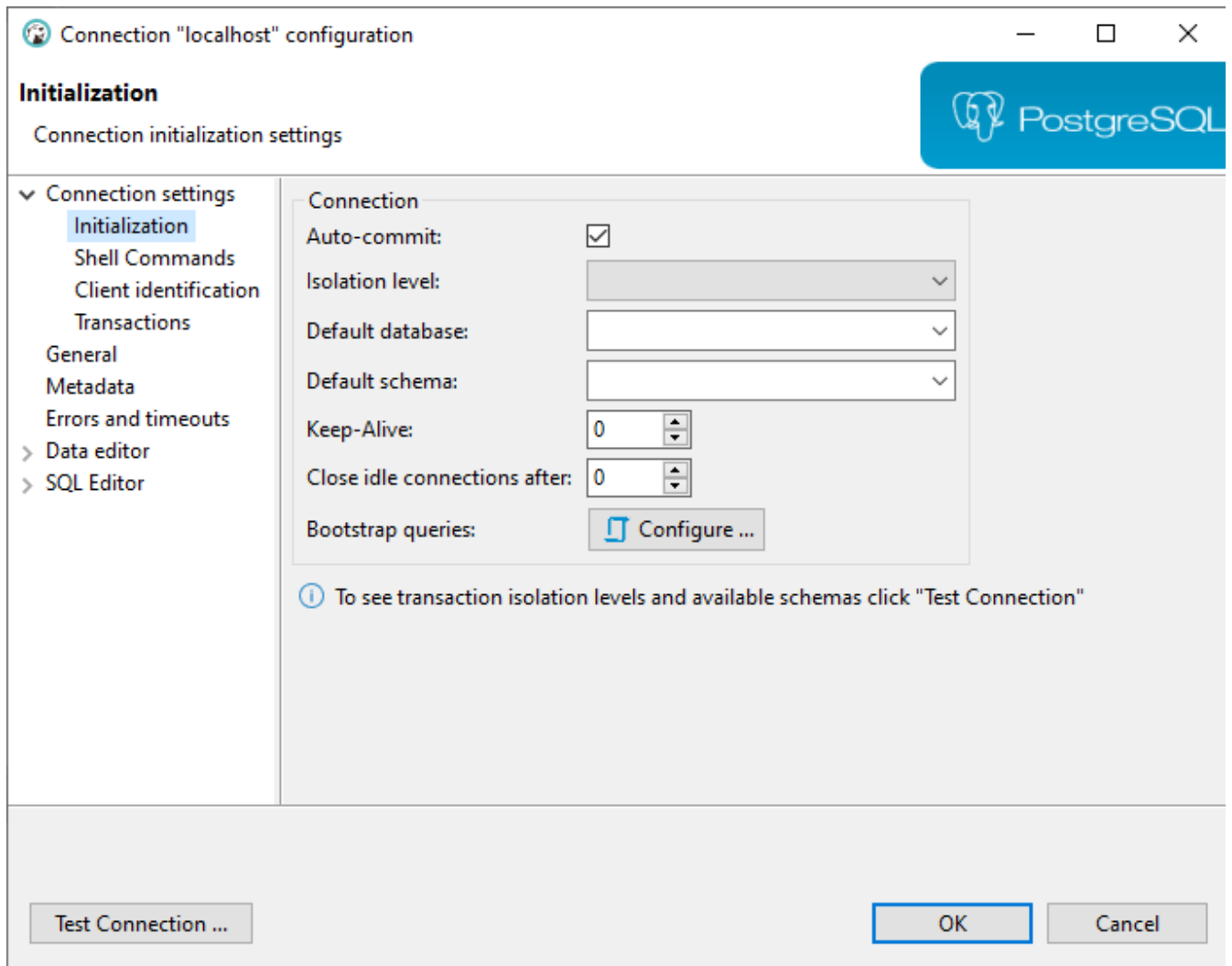
A jeżeli uda nam się poprawnie skonfigurować połączenie, zobaczymy ekran podobny do poniższego:



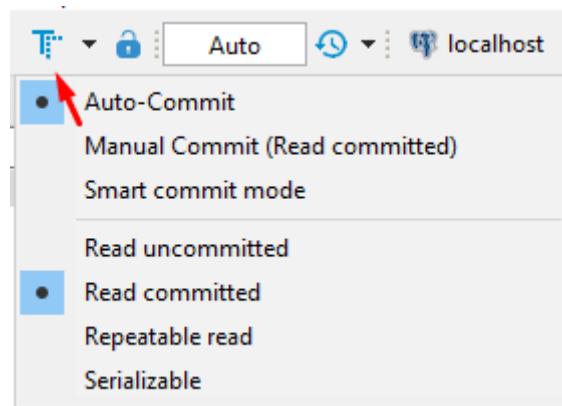
Jeżeli zamkniemy okno do dodania połączenia, i będziemy chcieli je skonfigurować ponownie albo coś zmienić, możemy przejść do ustawień przez prawy przycisk myszy i wybranie "Edit Connection":



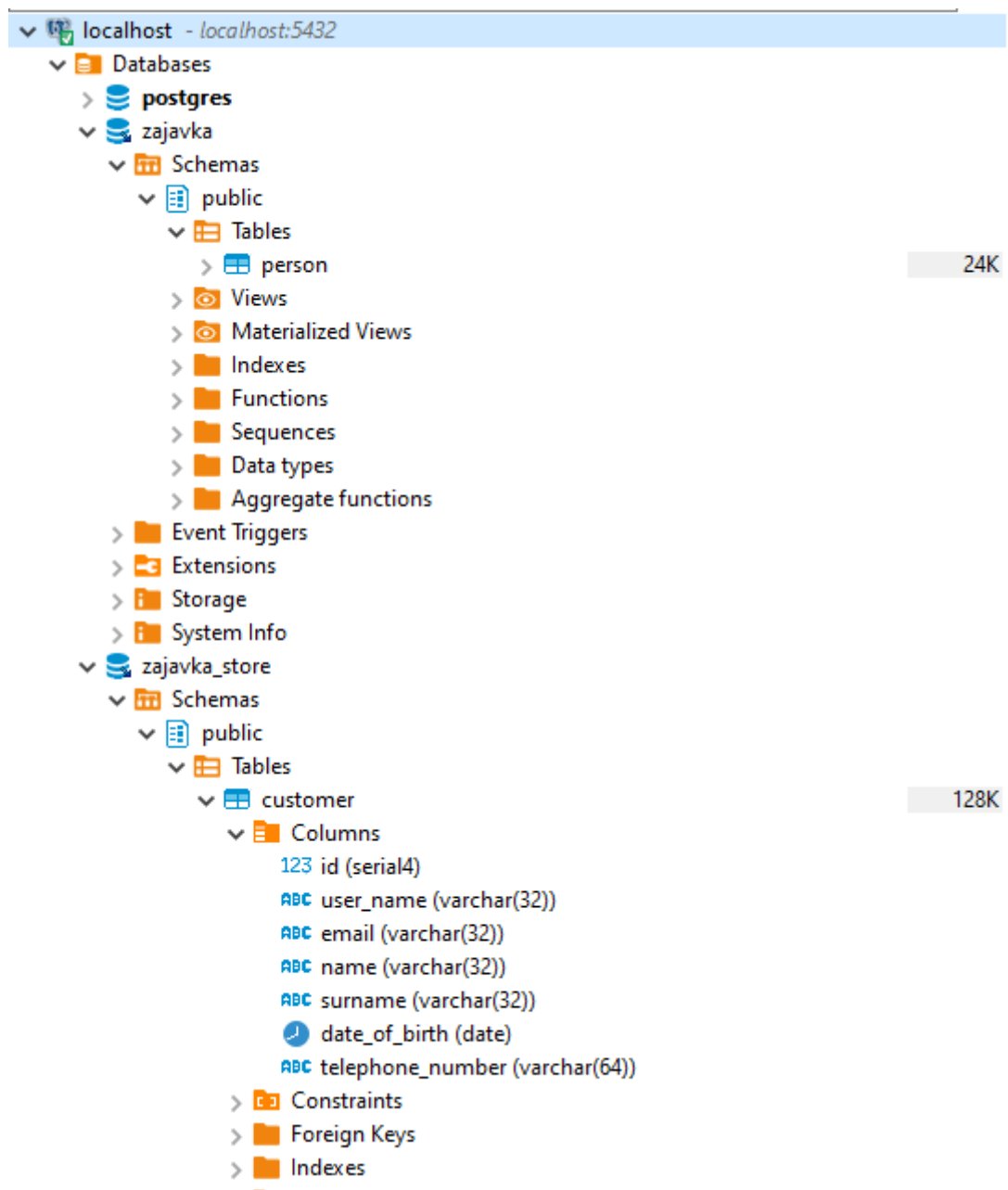
Jeżeli chodzi o transakcje, należy pamiętać o pewnej kwestii. Domyślnie nasze transakcje są commitowane automatycznie.



Możemy to wyłączyć również dzięki ikonie przypominającej literkę 'T'. Widać tutaj również możliwość przestawienia poziomu izolacji transakcji.



Gdy zaczniemy rozwijać bazy danych, do których mamy dostęp, zobaczymy dużo informacji, które są nam potrzebne do pracy z takimi bazami.



I wisienka na torcie. Żeby z poziomu tego narzędzia wykonywać zapytania, należy wybrać bazę danych i wybrać przycisk "F3". Będziemy wtedy mogli wpisywać zapytania SQL, a same wykonanie zapytania jest możliwe przy wykorzystaniu skrótu "ctrl + enter".



DBBeaver 22.1.1 - <localhost> Script

File Edit Navigate Search SQL Editor Database Window Help

SQL Commit Rollback Auto localhost public@zajavka

Database Navigator Projects

Enter a part of object name here

localhost - localhost:5432

- Databases
  - postgres
    - zajavka
      - Schemas
        - public
          - Tables
            - person 24K
          - Views
          - Materialized Views
          - Indexes
          - Functions
          - Sequences
          - Data types
          - Aggregate functions
        - Event Triggers
        - Extensions
        - Storage
        - System Info

Project - General

Name DataSource

- Bookmarks
- ER Diagrams
- Scripts

\*<localhost> Script

```
select * from person;
```

person 1

select \* from person Enter a SQL expression to filter results (use C

	id	name	age
1	12	Karol	15
2	15	Stefan	55
3	32	Karol	56
4	51	Karol	73
5	61	Karol	12