

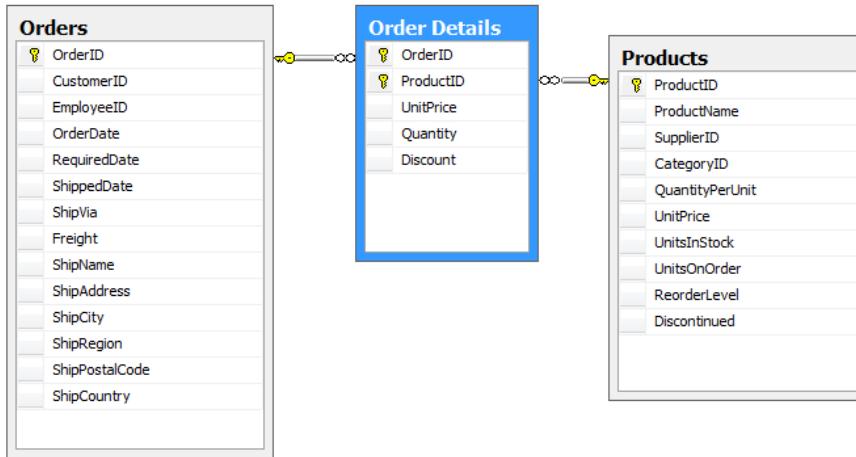
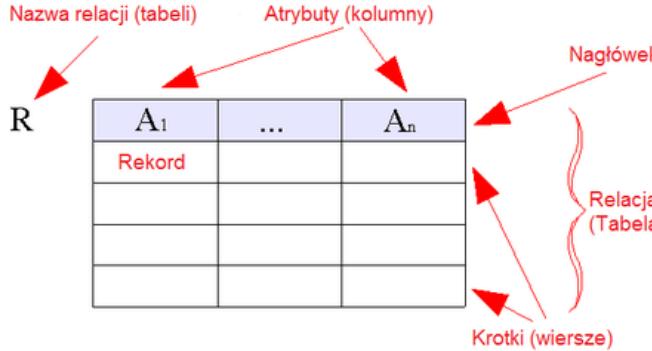
Relacyjna i nierelacyjna baza danych



RELACYJNA BAZA DANYCH

Relacyjne bazy danych przechowują i organizują punkty danych ze zdefiniowanymi relacjami w celu szybkiego dostępu. Dane są zorganizowane w tabelach, które przechowują informacje o każdej encji i reprezentują wstępnie zdefiniowane kategorie za pośrednictwem wierszy i kolumn. Do zarządzania i tworzenia zapytań używa się SQL, który jest w tym przypadku najpopularniejszym językiem.

Kiedy dokonujesz rezerwacji w hotelu, na pokładzie samolotu twoje dane są zapisywane w relacyjnej bazie danych. System rezerwacji przypisuje dostępność do konkretnych miejsc, zapewnia spójność danych i umożliwia szybkie przeszukiwanie dostępnych opcji.



Przykład działania - relacyjna baza danych

np.w środowisku implementacyjnym MS Access - system z zamówieniami internetowymi

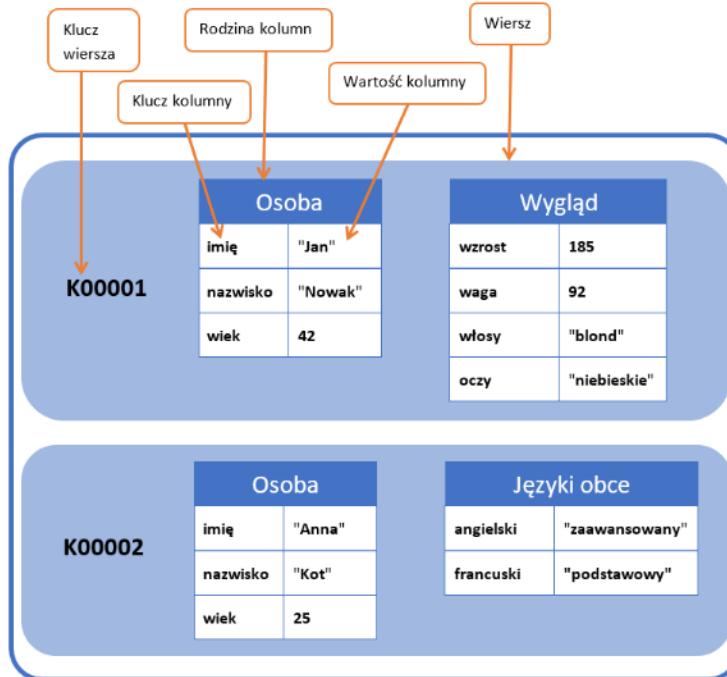
Działa, przechowując dane w formie tabel, a kolumny w tabelach reprezentują atrybuty tych danych. Dane są przechowywane w postaci rekordów w tych tabelach, a klucze główne jednoznacznie identyfikują każdy rekord. Operacje na danych wykonywane są za pomocą języka zapytań SQL.

NIERELACYJNA BAZA DANYCH

Nierelacyjna baza danych (*Not Only SQL*) nie ma sztywnego schematu danych, jest elastyczna w przechowywaniu różnorodnych typów oraz zazwyczaj nie używa języka zapytań SQL do operacji. Projektowane są w celu obsługi dużych ilości danych, a także cechują się łatwością w dostosowywaniu się do zmiennych potrzeb projektu.

Aplikacje społecznościowe i media społecznościowe używają nierelacyjnych baz danych. Dzięki nim przechowują i zarządzają ogromnymi ilościami danych użytkowników (posty, komentarze, polubienia i duża ilość multimedialnych treści)

NIERELACYJNE BAZY DANYCH



Można podzielić je na kilka kategorii pod względem modelu danych.

W przypadku baz rodziny kolumn, jak sama nazwa wskazuje, dane przechowywane są w formie rodziny kolumn - grupy spokrewnionych danych, które zazwyczaj pobierane są razem. Te z kolei, przechowywane są w wierszach z przypisanymi kluczami.

Koncepcję tego typu bazy danych można przyrównać do ogromnej, zdemoralizowanej tabeli liczącej wiele wierszy i kolumn.

PODSTAWOWE RÓŻNICE



	RELACYJNA BAZA DANYCH	NIERELACYJNA BAZA DANYCH
STRUKTURA	Tabele o określonej liczbie kolumn	Dokumentowe, grafowe, kolumnowe lub klucz-wartość
RODZAJ STRUKTURY DANYCH	Stała struktura danych	Zmienna struktura danych
ZAPYTANIA	Przeważnie SQL	Nie ma deklaratywnego języka zapytań
SKALOWANIE	Skalowanie w pionie – zwiększa obciążenie serwera	Skalowanie w poziomie dzieląc je na fragmenty między serwerami
SCENARIUSZE	Księgowość, finanse, bankowość, zarządzanie transakcjami,	Aplikacje mobilne, personalizacja, migracja baz danych, zarządzanie zawartością
NAJLEPSZA DLA:	-Obsługa relacyjnych danych mających logiczne i odrębne wymagania, które można określić z góry. -Schemat, który wymaga konserwacji oraz synchronizacji aplikacji i bazy danych. -Aplikacje wymagające skomplikowanych zapytań lub transakcji wielowierszowych.	-Obsługa dużych ilości niepowiązanych, nieokreślonych lub szybko zmieniających się danych. -Dane o nieznanym schemacie lub schemacie podyktyowanym przez aplikację. -Aplikacje, w przypadku których wydajność i dostępność są ważniejsze od silnej spójności. -Zawsze włączone aplikacje służące użytkownikom na całym świecie.
NAJWAŻNIEJSZE CECHY	Spójne i stabilne dane	Elastyczne i szybkie zapytania

DZIĘKUJĘ ZA UWAGĘ!

Ilona Batko

BIBLIOGRAFIA

<https://www.oracle.com/pl/database/what-is-a-relational-database/> [dostęp z nia 06.11.2023]

<https://azure.microsoft.com/pl-pl/resources/cloud-computing-dictionary/what-is-a-relational-database> [dostęp z nia 06.11.2023]

<https://azure.microsoft.com/pl-pl/resources/cloud-computing-dictionary/what-is-nosql-database>
[dostęp z dnia 06.11.2023]

<https://inetum.pl/nosql-vs-sql-bazy-danych/> [dostęp z dnia 06.11.2023]

<https://edu.pjwstk.edu.pl/wyklady/rbd/scb/wyklad9/sql.htm> [dostęp z dnia 06.11.2023]

<https://b2bnetwork.pl/zwiazek-bez-relacji-nierelacyjne-bazy-danych-cz-2/> [dostęp z dnia 07.11.2023]

<http://www.informatyka.zs6.szczecin.pl/20.html> [dostęp z dnia 07.11.2023]

<https://itwiz.pl/czym-jest-nosql-jak-wykorzystac-nierelacyjne-bazy-danych/> [dostęp z dnia 07.11.2023]

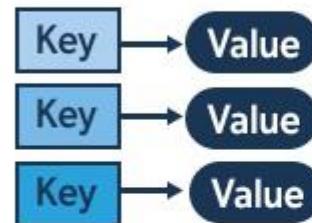
Nierelacyjna baza danych - podstawowe pojęcia i ich znaczenie

Co to jest nierelacyjna baza danych?

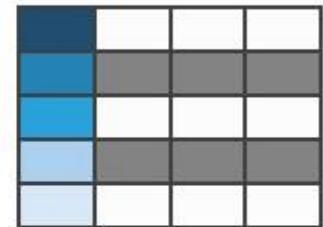
Nierelacyjne bazy danych (NoSQL) to rodzaj systemów zarządzania danymi, które różnią się od tradycyjnych baz danych relacyjnych pod wieloma względami. Główną różnicą jest brak ścisłe zdefiniowanych schematów i struktur tabel. W nierelacyjnych bazach danych dane są przechowywane w bardziej elastyczny sposób, co pozwala szybkie i skalowalne operacje

NoSQL

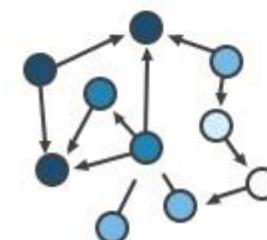
klucz-wartość



kolumnowe



grafowe



dokumentowe



Typy nierelacyjnych baz danych:

1

Bazy danych klucz-wartość Dane są przechowywane w postaci par klucz-wartość, co pozwala na szybkie dostępy do informacji.

2

Bazy danych kolumnowe Dane są przechowywane w kolumnach, co pozwala na efektywne analizowanie dużych ilości danych.

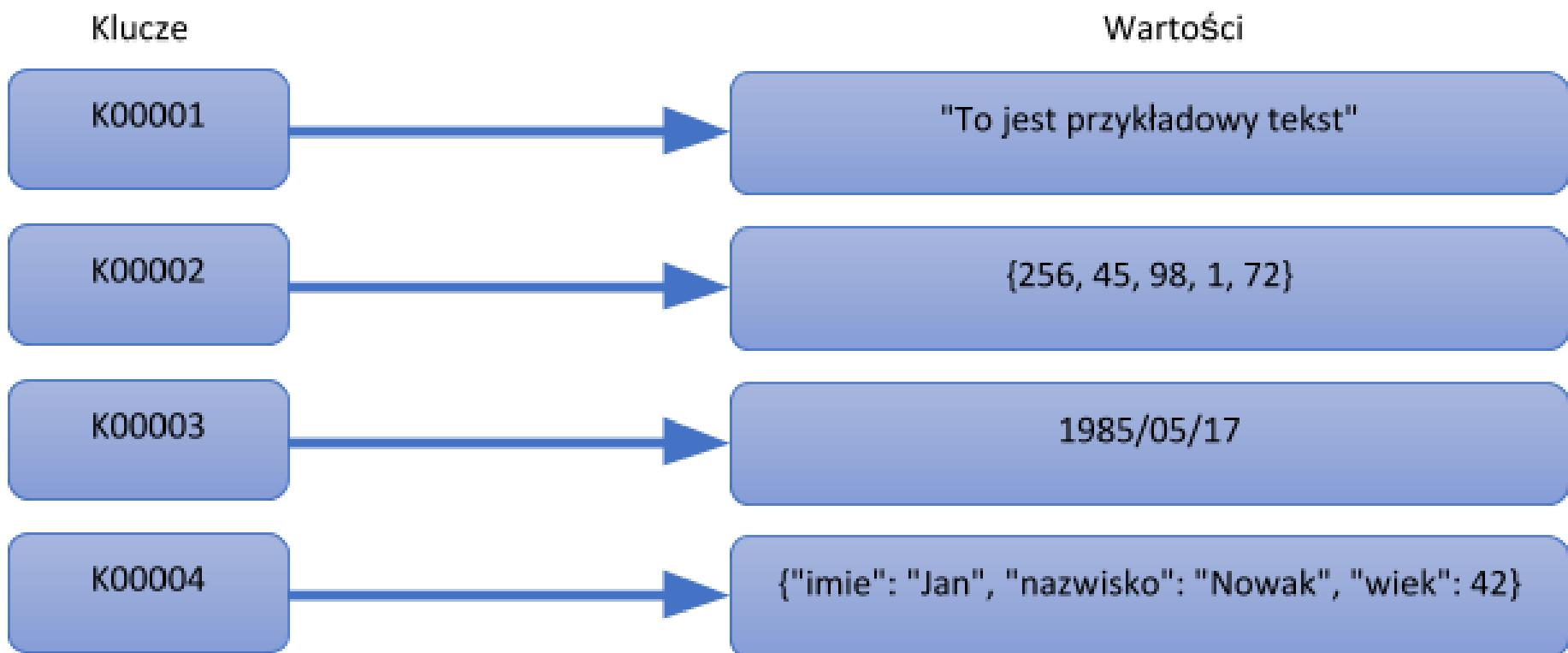
3

Bazy danych grafowe Służą do przechowywania danych w formie grafów, co jest szczególnie przydatne do analizy związków między danymi.

4

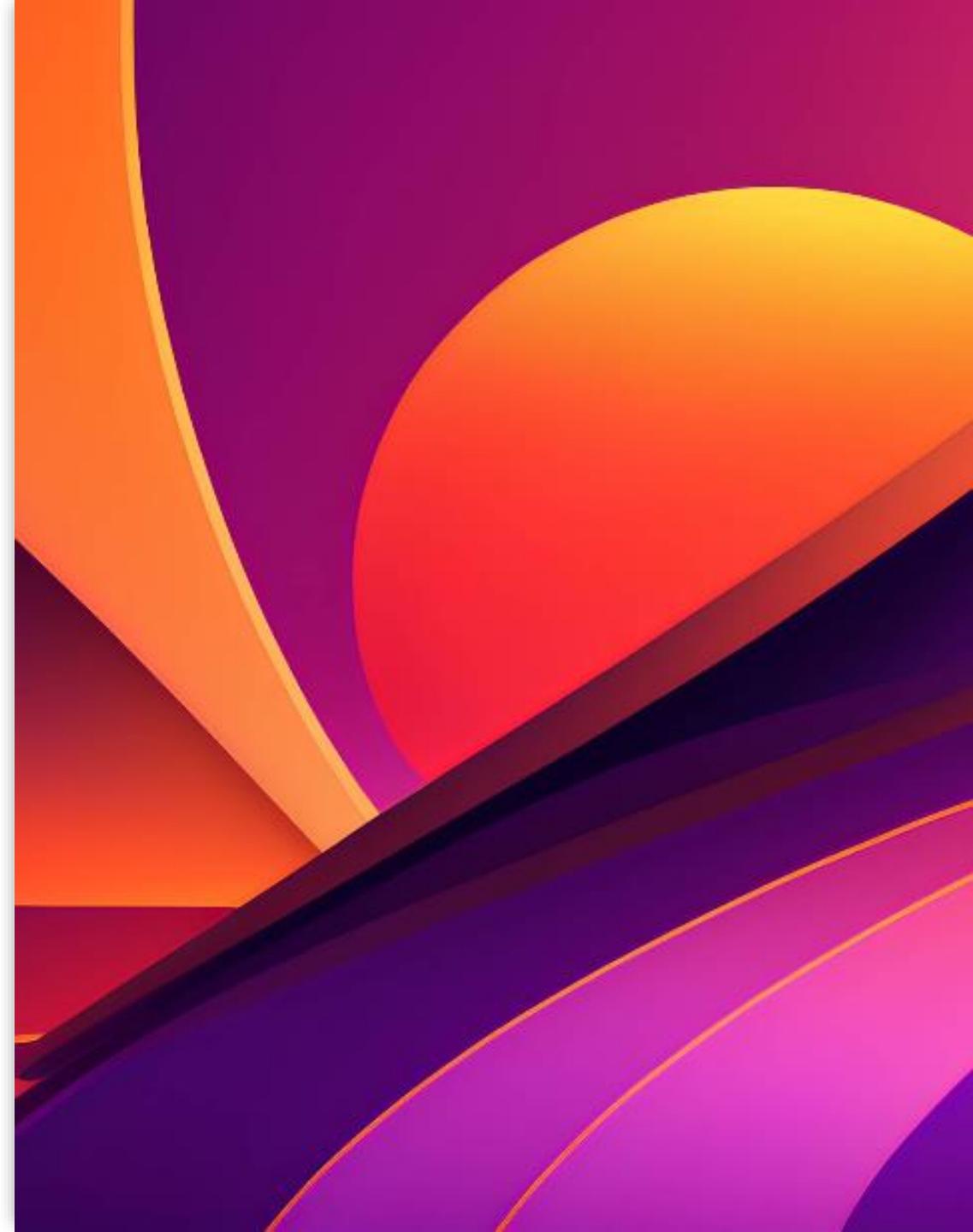
Bazy danych dokumentowe Przechowują dane w postaci dokumentów, często w formacie JSON.

Przykład bazy danych klucz wartość



Znaczenie nierelacyjnych baz danych

- **Skalowalność:** Nierelacyjne bazy danych są łatwiejsze do skalowania w przypadku wzrostu ilości danych lub obciążenia.
- **Elastyczność:** Brak ściśle określonych schematów pozwala na zmiany w strukturze danych bez konieczności przebudowy całej bazy.
- **Szybkość:** Dzięki specyficzny strukturom przechowywania danych, wiele nierelacyjnych baz danych oferuje wyjątkowo szybkie operacje odczytu i zapisu.
- **Zastosowania specjalistyczne:** Poszczególne rodzaje nierelacyjnych baz danych są dostosowane do różnych zastosowań, co pozwala na efektywne przechowywanie różnych typów danych.



Przykłady nierelacyjnych baz danych



Nierelacyjna baza danych wykorzystująca strukturę danych typu klucz-wartość.

Jedna z najpopularniejszych nierelacyjnych baz danych, oparta na modelu dokumentowym.

Rozproszona nierelacyjna baza danych o wysokiej dostępności i skalowalności.



Podsumowanie

Nierelacyjne bazy danych stanowią ważne element dzisiejszego ekosystemu danych i technologii informacyjnych. Ich elastyczność, skalowalność i szybkość przyczyniają się do efektywnego zarządzania ogromnymi ilościami informacji, które generujemy każdego dnia. Niemniej jednak, wybór rodzaju baz danych zależy od konkretnych potrzeb i charakterystyki problemu, co sprawia że zarówno bazy relacyjne, jak i nie relacyjne mają swoje miejsce w dzisiejszym świecie informatyki.



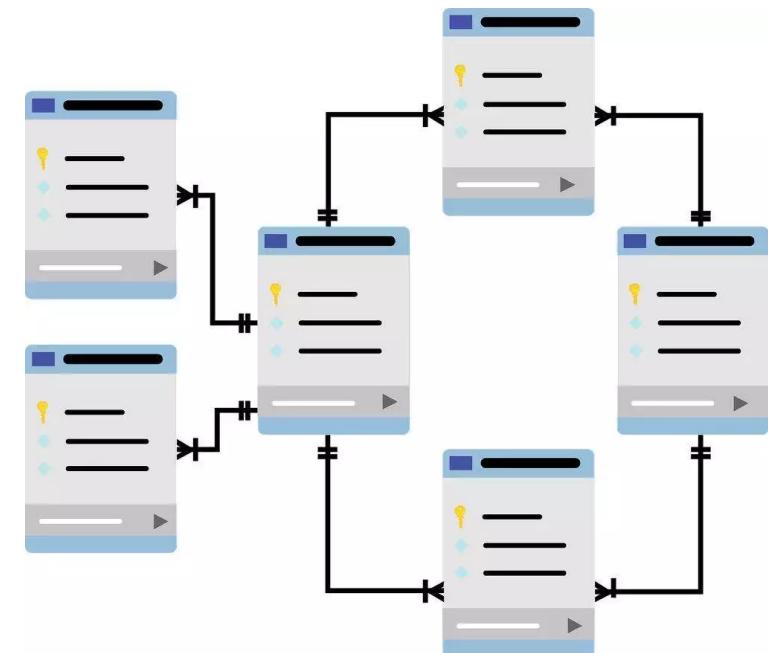
Bibliografia

- ▶ <https://www.youtube.com/watch?v=78EOiGTTtQI> [dostęp 7.11.23]
- ▶ <https://www.youtube.com/watch?v=qh7d6Xf0KGk> [dostęp 7.11.23]
- ▶ <https://tworcastron.pl/blog/sql-vs-nosql-mysql-vs-mongodb-roznice/> [dostęp 7.11.23]
- ▶ <https://www.oracle.com/pl/database/nosql/what-is-nosql/> [dostęp 7.11.23]
- ▶ <https://inetum.pl/nosql-vs-sql-bazy-danych/> [dostęp 7.11.23]
- ▶ <https://tworcastron.pl/blog/sql-vs-nosql-mysql-vs-mongodb-roznice/> [dostęp 7.11.23]
- ▶ <https://itwiz.pl/czym-jest-nosql-jak-wykorzystac-nierelacyjne-bazy-danych/> [dostęp 7.11.23]

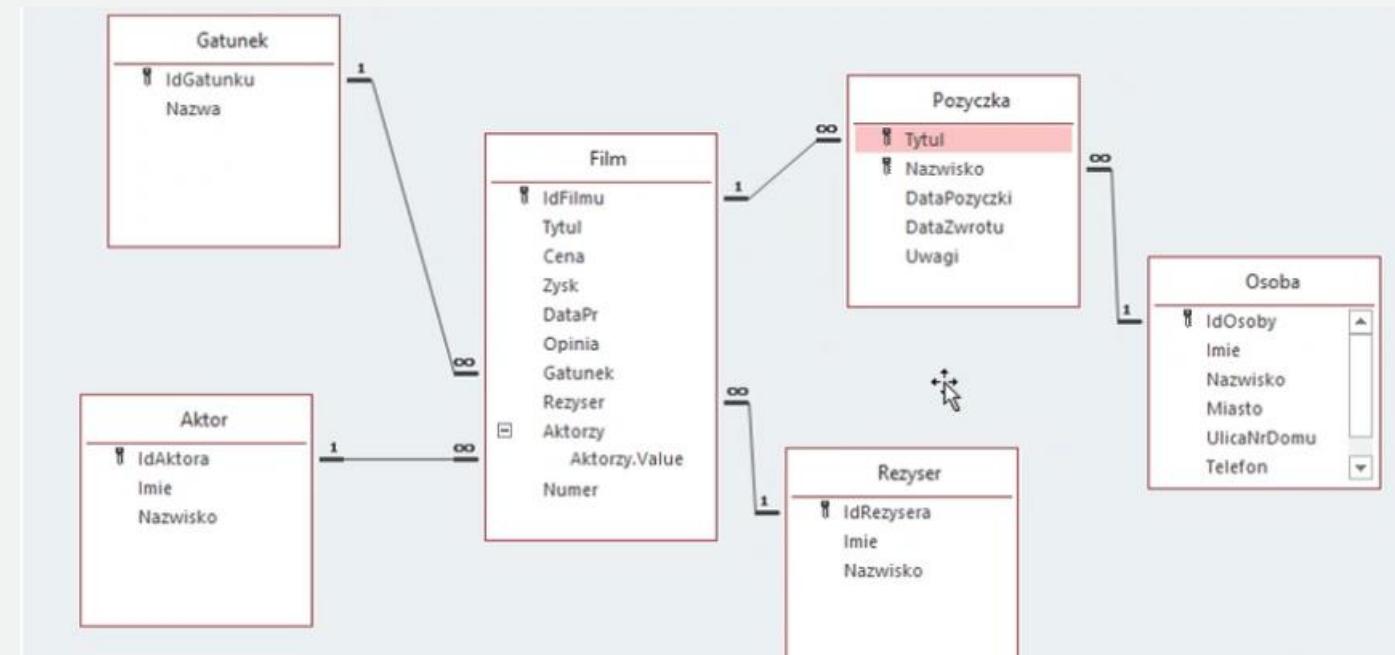
RELACYJNA BAZA DANYCH

ANETA ANTOS
ETI 3ROK

Rodzaj bazy danych, który pozwala przechowywać powiązane ze sobą elementy danych i zapewnia do nich dostęp.



Relacyjne bazy danych są oparte na modelu relacyjnym. W relacyjnej bazie danych każdy wiersz tabeli jest rekordem z unikatowym identyfikatorem nazywanym kluczem. Kolumny tabeli zawierają atrybuty danych, a każdy rekord zawiera zwykle wartość dla każdego atrybutu, co ułatwia ustalenie relacji między poszczególnymi elementami rekordu.



Standardowy język zapytań (SQL) to standardowy interfejs dla relacyjnych baz danych

Relacyjne bazy danych mają wysoką strukturę i są reprezentowane przy użyciu schematu (logicznego i fizycznego)

Relacyjne bazy danych zmniejszają ich nadmiarowość

Relacyjne bazy danych składają się z wielu jednostek



RELACYJNA BAZA DANYCH MYSQL

My Structured Query Language (MySQL) to wspólna relacyjna baza danych SQL typu open source, która wykonuje wszystkie podstawowe polecenia SQL, takie jak zapisywania i wykonywanie zapytań dotyczących danych. Szeroko wdrażany MySQL jest niezawodnym, stabilnym i bezpiecznym systemem zarządzania bazami danych, ponieważ obsługuje większość wiodących języków programowania i protokołów. Relacyjne bazy danych MySQL zwykle zawierają wzmacnione i elastyczne funkcje zabezpieczeń, takie jak weryfikacja oparta na hoście i ruch zaszyfrowany hasłem.

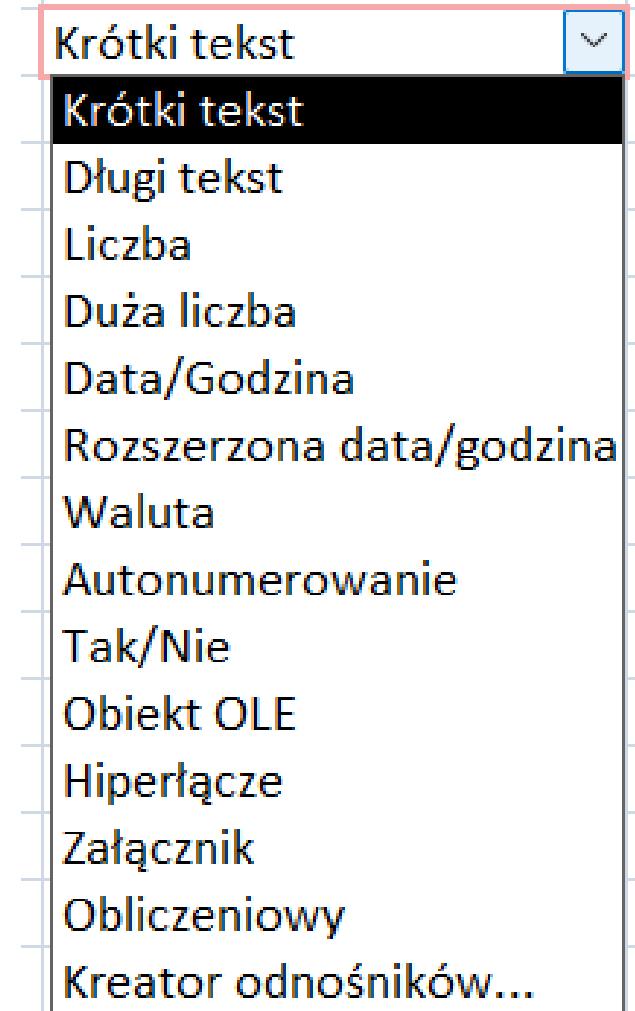
Z czego się składa?

Tabela (Table): Tabela jest podstawową strukturą w relacyjnej bazie danych. Reprezentuje ona zbiór rekordów lub wierszy, gdzie każdy wiersz zawiera informacje na temat konkretnej jednostki danych, a kolumny określają różne atrybuty tych jednostek danych.

Rekord (Row): Rekord to pojedynczy wiersz w tabeli. Reprezentuje konkretną jednostkę danych lub obiekt, a każda kolumna tabeli przechowuje wartości atrybutów tej jednostki danych.

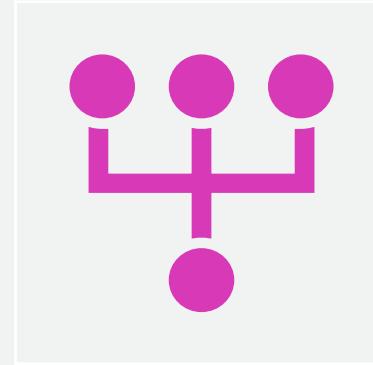
Kolumna (Column): Kolumna to atrybut lub pole w tabeli, które zawiera konkretne informacje o jednostce danych. Kolumny określają rodzaj danych, który jest przechowywany w danej kolumnie (np. tekst, liczba całkowita, data).

Klucz główny (Primary Key): Klucz główny to unikalny identyfikator każdego rekordu w tabeli. Zapewnia on jednoznaczna identyfikację rekordów i zapobiega powtarzaniu się tych samych danych w tabeli. Klucz główny jest zazwyczaj jednym z pól tabeli.





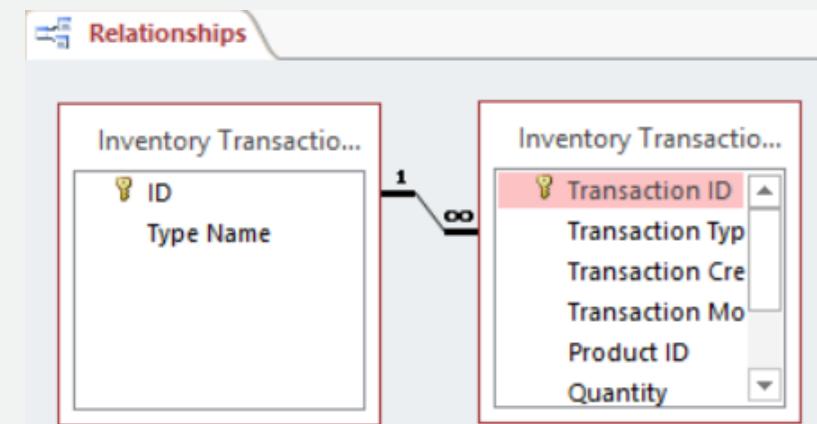
Klucz obcy (Foreign Key): Klucz obcy to pole w tabeli, które odnosi się do klucza głównego w innej tabeli. Służy do tworzenia relacji między różnymi tabelami w bazie danych, umożliwiając łączenie danych z różnych tabel.



Relacja (Relationship): Relacja jest powiązaniem między dwiema lub więcej tabelami w bazie danych. Jest to związek, który określa, jak rekordy z jednej tabeli są powiązane z rekordami w innej tabeli.

RELACJA JEDEN DO WIELU

- W tego rodzaju relacji wiersz w tabeli A może mieć wiele pasujących wierszy w tabeli B. Jednak wiersz w tabeli B może mieć tylko jeden pasujący wiersz w tabeli A.
- Relacja jeden-do-wielu jest tworzona, jeśli tylko jedna z powiązanych kolumn jest kluczem podstawowym lub ma unikatowe ograniczenie.
- W oknie relacji w programie Access strona klucza podstawowego relacji jeden-do-wielu jest oznaczona przez liczbę 1. Strona klucza obcego relacji jest oznaczona symbolem nieskończoności.



RELACJA WIELE DO WIELU

- Wiersz w tabeli A może mieć wiele pasujących wierszy w tabeli B i odwrotnie. Taką relację można utworzyć, definiując trzecią tabelę, która jest nazywana tabelą skrzyżowań. Klucz podstawowy tabeli skrzyżowań składa się z kluczy obcych z tabeli A i tabeli B. Na przykład tabela „Autorzy” i tabela „Tytuły” są w relacji wiele-do-wielu, która jest definiowana przez relację jeden-do-wielu z każdej z tych tabel do tabeli „TytułAutorzy”.



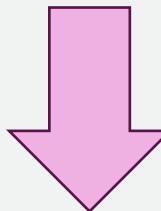
RELACJA JEDEN DO JEDNEGO

- W relacji jeden-do-jednego wiersz w tabeli A może mieć nie więcej niż jeden pasujący wiersz w tabeli B i odwrotnie. Relacja jeden-do-jednego jest tworzona, jeśli obie kolumny powiązane są kluczami podstawowymi lub mają wyjątkowe ograniczenia.
- Ten rodzaj relacji nie jest powszechny, ponieważ większość informacji, które są związane w ten sposób znajdzie się w jednej tabeli.

Relacji jeden-do-jednego można użyć do podjęcia następujących czynności:

- Podzielenia tabeli z wieloma kolumnami.
- Izolowania część tabeli ze względów bezpieczeństwa.
- Przechowywania danych, które są krótkotrwałe i można je łatwo usunąć, usuwając tabelę.
- Przechowywania informacji, które mają zastosowanie tylko do podzbioru tabeli głównej. W programie Access strona klucza podstawowego relacji jeden-do-jednego jest oznaczona symbolem klucza. Strona klucza obcego jest również oznaczona symbolem klucza.

Nazwisko	Imię	Telefon	Miasto	Zawód
Kowalski	Jan	12673452	Kraków	kierowca
Klich	Zbigniew	66589123	Warszawa	portier
Marchewka	Edyta	70145278	Kraków	nauczyciel
Siwak	Judyta	70588821	Rzeszów	nauczyciel
...



Nazwisko	Imię	Telefon	Miasto	Zawód
Kowalski	Jan	12673452	1	1
Klich	Zbigniew	66589123	2	2
Marchewka	Edyta	70145278	1	3
Siwak	Judyta	70588821	3	3

Miasto
Kraków
Warszawa
Rzeszów

1
2
3

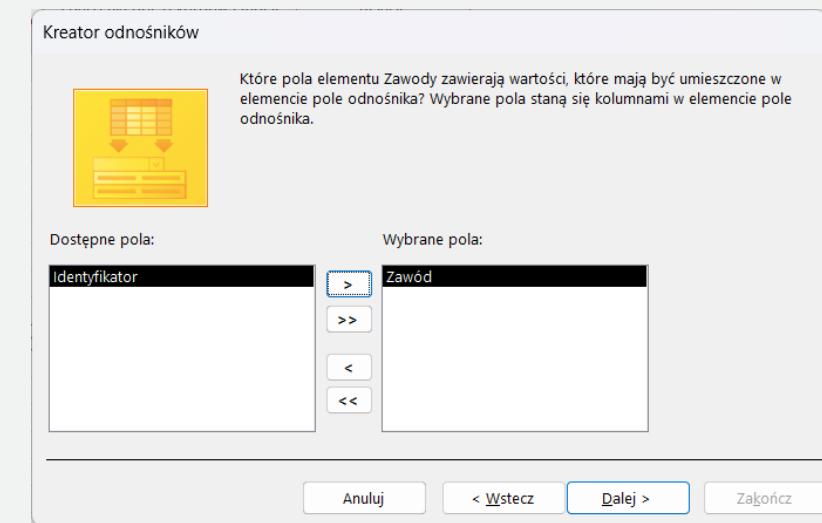
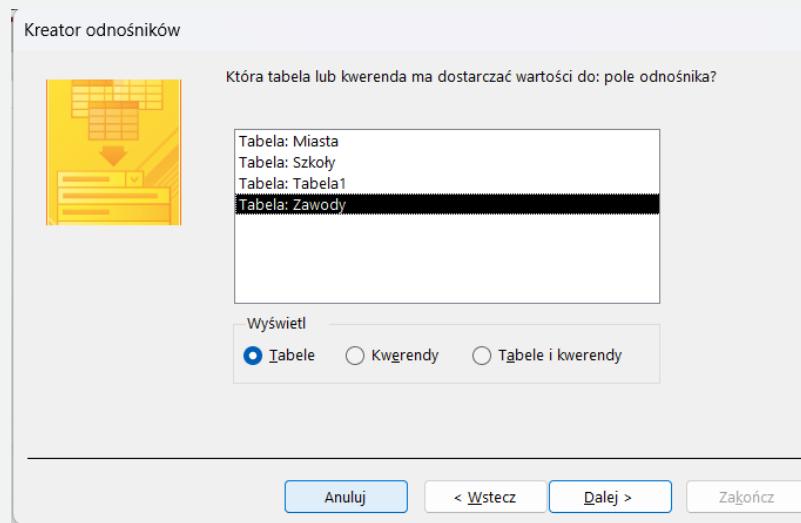
Zawód
kierowca
portier
nauczyciel

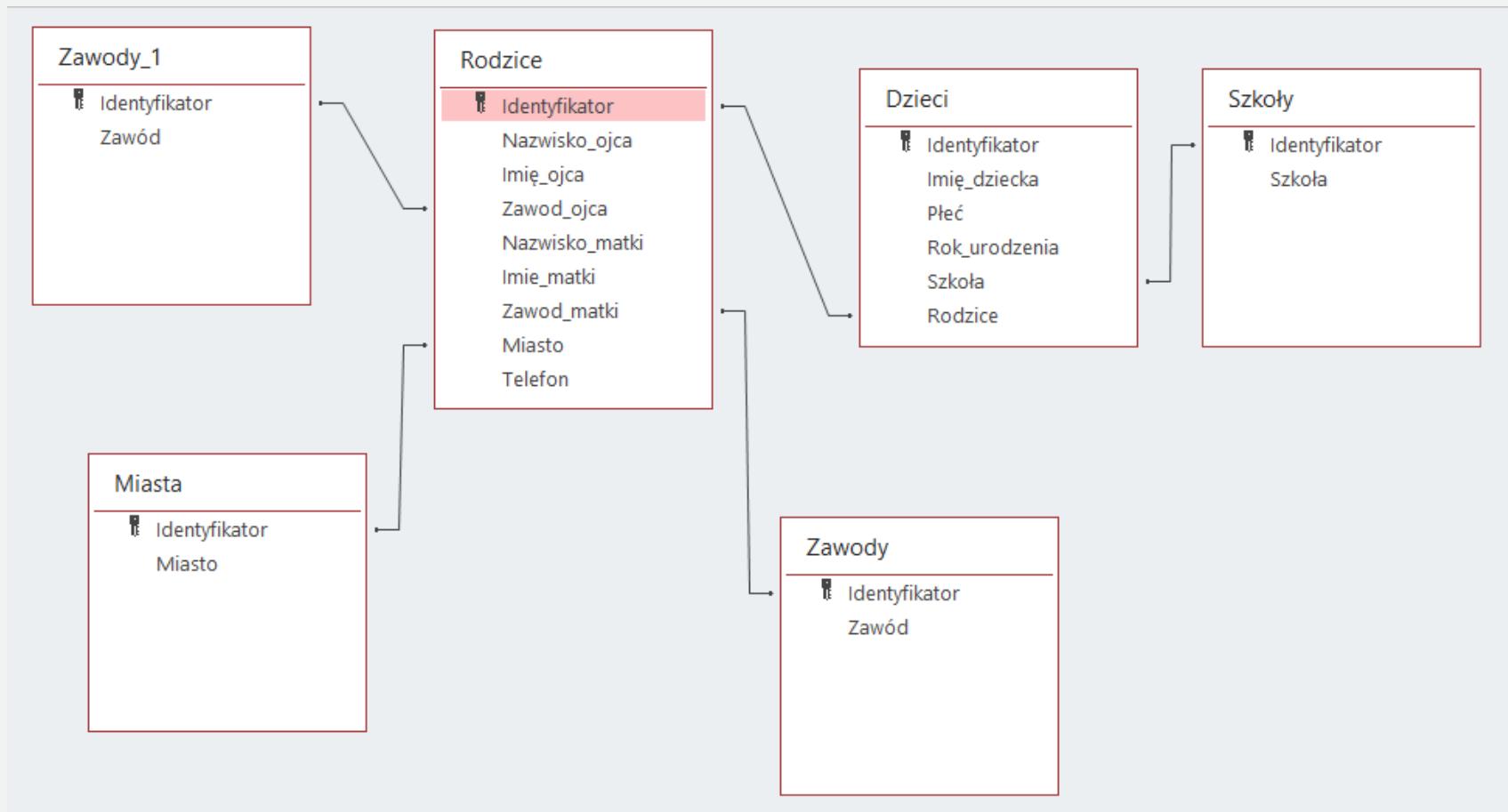
1
2
3

Identyfikator	Miasto	Kliknij, aby dodać
1	Tarnów	
2	Kraków	
3	Warszawa	
4	Rzeszów	
5	Bochnia	
6	Miechów	
7	Słomniki	
8	Poznań	
*	(Nowy)	

Identyfikator	Zawód	Kliknij, aby dodać
1	Kierownik	
2	Sekretarz	
3	Prezes	
4	Kierowca	
5	Portier	
6	Inżynier	
7	Nauczyciel	
*	(Nowy)	

Identyfikator	Szkoła	Kliknij, aby dodać
1	SP 1	
2	SP 2	
3	SP 3	
4	I LO	
5	II LO	
6	III LO	
7	IV LO	
8	Przedszkole	
*	(Nowy)	





Bibliografia:

- <https://azure.microsoft.com/pl-pl/resources/cloud-computing-dictionary/what-is-a-relational-database>
- <https://www.oracle.com/pl/database/what-is-a-relational-database/>
- eduinf.waw.pl
- <https://learn.microsoft.com/pl-pl/office/troubleshoot/access/define-table-relationships>
- <http://rafalorzelek.pl/bazy-danych/44-06-typy-relacji.html>

MONGODB – COMPASS/ATLAS

MAREK PALIŃSKI ETI III ROK

MONGODB - COMPASS

- MongoDB Compass to oficjalne narzędzie do zarządzania bazami danych MongoDB.



MONGODB - ATLAS

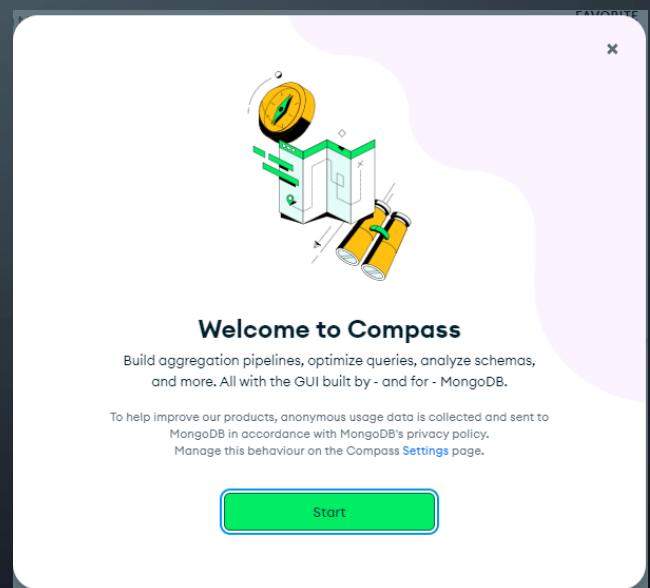
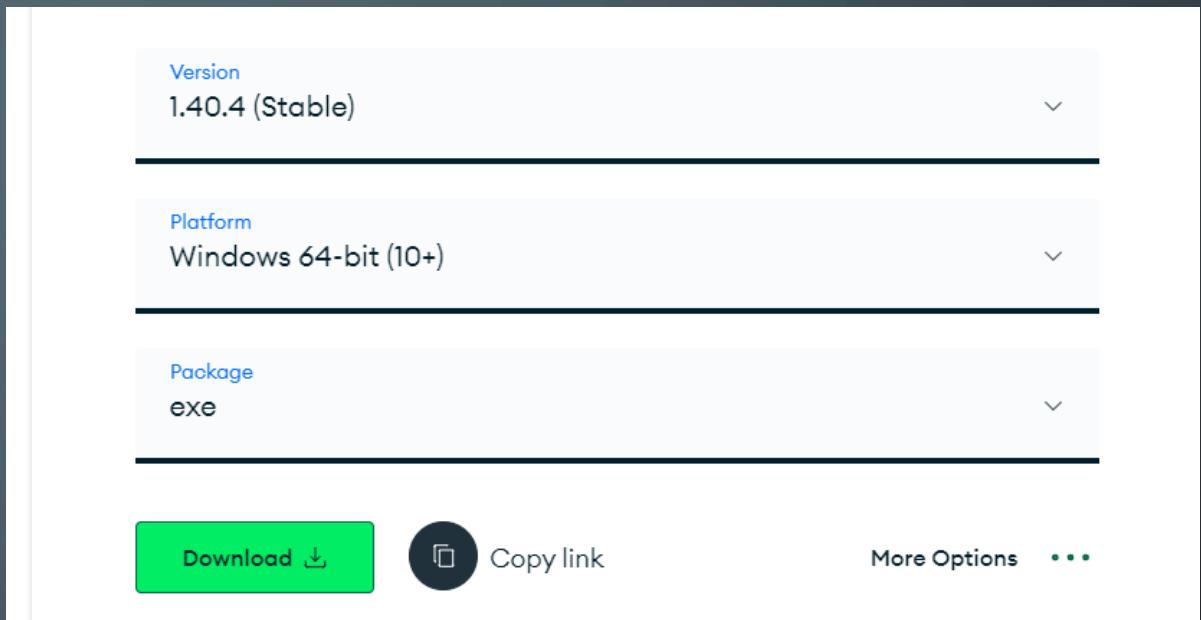
- MongoDB Atlas to usługa chmurowa, która oferuje zarządzane instancje MongoDB w chmurze.



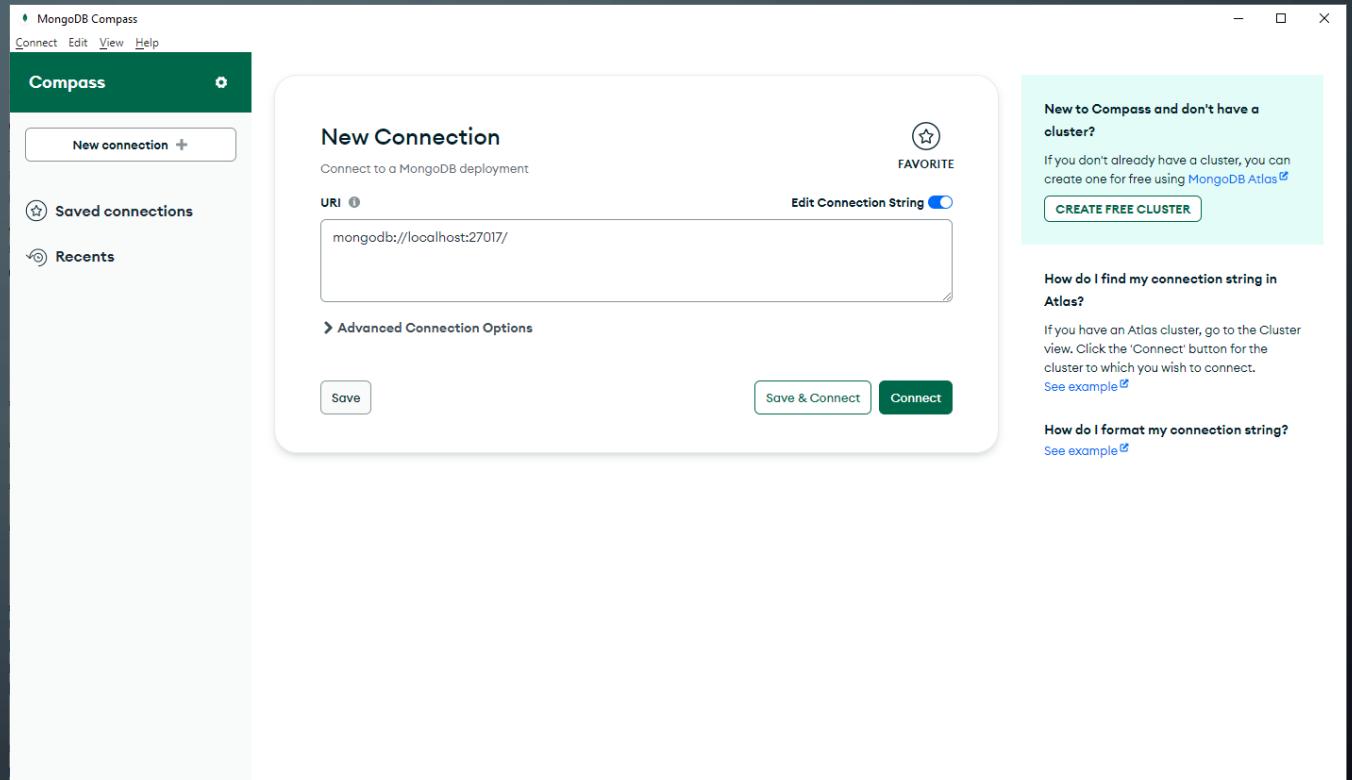
KROK PO KROKU

<https://www.mongodb.com/try/download/shell>

Instalujemy wersję zgodną z naszym systemem operacyjnym



KROK PO KROKU



KROK PO KROKU

MongoDB.

Deploy your database

Use a template below or set up [advanced configuration options](#). You can also edit these configuration options once the cluster is created.

M10	\$0.08/hour	
For production applications with sophisticated workload requirements.		
STORAGE 10 GB	RAM 2 GB	vCPU 2 vCPUs

SERVERLESS	\$0.10/1M reads	
For application development and testing, or workloads with variable traffic.		
STORAGE Up to 1TB	RAM Auto-scale	vCPU Auto-scale

M0	FREE	
For learning and exploring MongoDB in a cloud environment.		
STORAGE 512 MB	RAM Shared	vCPU Shared

Provider

Region

★ Recommended region [i](#)

N. Virginia (us-east-1) ★

Name

You cannot change the name once the cluster is created.

FREE

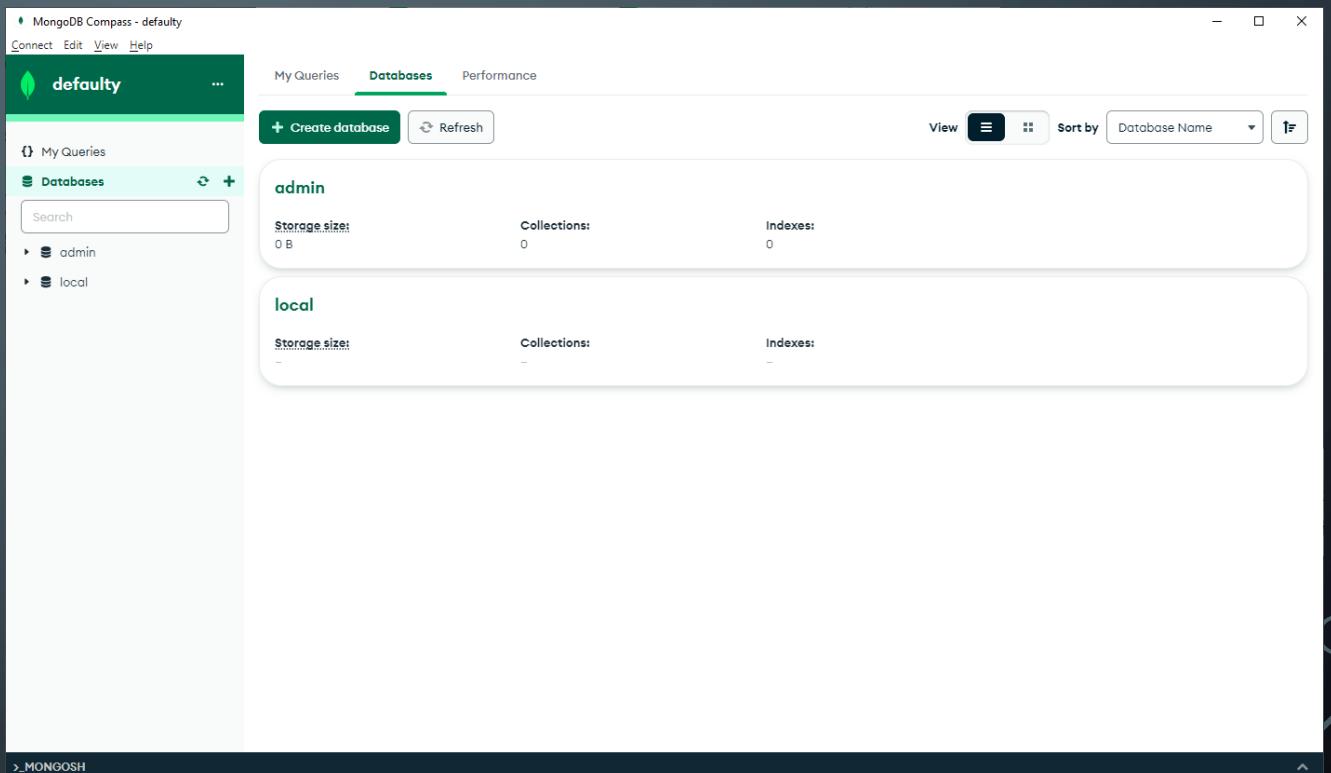
[Create](#)

[Access Advanced Configuration](#)

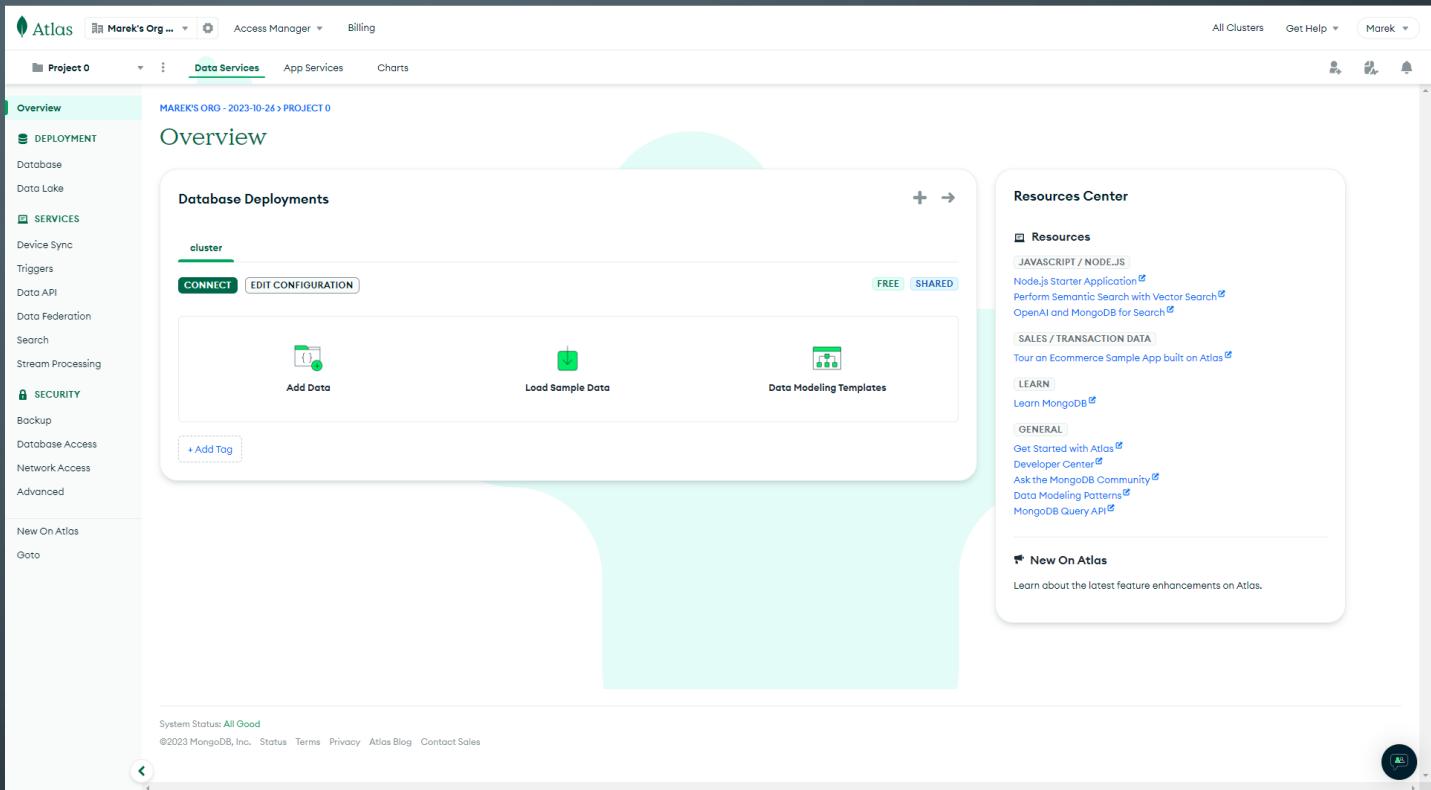
Free forever! Your M0 cluster is ideal for experimenting in a limited sandbox. You can upgrade to a production cluster anytime.

I'll deploy my database later

KROK PO KROKU



KROK PO KROKU



KROK PO KROKU

Connect to cluster



Connecting with MongoDB Compass

[I don't have MongoDB Compass installed](#)

[I have MongoDB Compass installed](#)

1. Choose your version of Compass

1.12 or later

See your Compass version in "About Compass"

2. Copy the connection string, then open MongoDB Compass

`mongodb+srv://marek182003db:<password>@cluster.tonpeo.mongodb.net/`



Replace `<password>` with the password for the `marek182003db` user.

When entering your password, make sure that any special characters are [URL encoded](#).

RESOURCES

[Connect with Compass](#)

[Access your Database Users](#)

[Import and Export Data](#)

[Troubleshoot Connections](#)

[Go Back](#)

[Close](#)

PRZESTAWIENIE DZIAŁANIA

MongoDB Compass - default/Sklep

Connect Edit View Help

default

My Queries

Databases

Search

Sklep

flowers

admin

local

Collections

+ Create collection Refresh

View

Sort by Collection Name

flowers

Storage size: 20.48 kB

Documents: 1

Avg. document size: 67.00 B

Indexes: 1

Total index size: 20.48 kB

_MONGOSH

This screenshot shows the MongoDB Compass interface. The left sidebar displays the database structure with 'Sklep' selected. The main area shows the 'Collections' tab with one collection named 'flowers'. Below the collection name, its storage statistics are listed: Storage size: 20.48 kB, Documents: 1, Avg. document size: 67.00 B, Indexes: 1, and Total index size: 20.48 kB. The document itself is not visible in the preview pane.

COMPASS

MongoDB Compass - default/Sklep.flowers

Connect Edit View Collection Help

default ...

Documents Sklep.flowers +

My Queries Databases Search

Sklep flowers ...

admin local

Sklep.flowers

1 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or [Generate query](#) Explain Reset Find Options

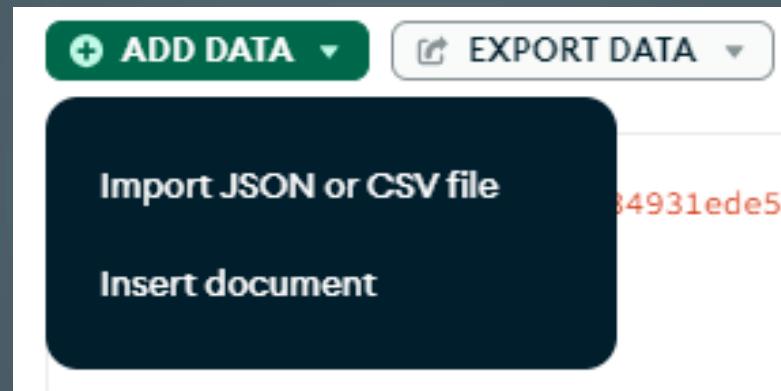
ADD DATA EXPORT DATA 1-1 of 1

```
_id: ObjectId('654934931ede5d9395c4db53')
type: "roses"
colour: "blue"
amount: 124
```

>_MONGOSH

This screenshot shows the MongoDB Compass interface. The left sidebar displays databases (default, Sklep, admin, local) and collections (flowers). The main area shows the 'Sklep.flowers' collection with one document. The document details are: _id: ObjectId('654934931ede5d9395c4db53'), type: "roses", colour: "blue", amount: 124. The top right corner indicates 1 DOCUMENTS and 1 INDEXES. The bottom status bar shows the prompt >_MONGOSH.

COMPASS



INSERT DOCUMENT

Insert Document

To collection Sklep.flowers

VIEW () ≡

```
1  /**
2   * Paste one or more documents here
3   */
4  {
5    "_id": {
6      "$oid": "654cb852c629ac3b035717fa"
7    }
8 }
```

Cancel Insert

IMPORT JSON OR CSV

Import

To collection Sklep.flowers

Import file: strong779791079407882002.csv edit

Options

Select delimiter Semicolon ▼

Ignore empty strings

Stop on errors

Specify Fields and Types [Learn more about data types](#)

Date	Workout Name	Exercise Name	Set Order	Weight
<input checked="" type="checkbox"/> String	<input checked="" type="checkbox"/> String	<input checked="" type="checkbox"/> String	<input checked="" type="checkbox"/> Int32	<input checked="" type="checkbox"/> Mixed
1 2023-02-20 13:16:17	Midday Workout	Bench Press (Dumbbell)	1	7,5
2 2023-02-20 13:16:17	Midday Workout	Bench Press (Dumbbell)	2	10

Cancel Import

Create Collection

x

Collection Name

|

Time-Series

Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

› **Additional preferences** (e.g. Custom collation, Capped, Clustered collections)

Cancel

Create Collection

COMPASS

NODE.JS

Node.js

```
const MongoClient = require('mongodb').MongoClient;

// Połączenie do serwera MongoDB
const url = 'mongodb://localhost:27017';
const dbName = 'sklepKwiatowy';

MongoClient.connect(url, { useNewUrlParser: true }, (err, client) => {
  if (err) {
    console.error('Błąd podczas połączenia z bazą danych:', err);
    return;
  }

  // Utworzenie bazy danych "sklepKwiatowy"
  const db = client.db(dbName);

  // Utworzenie kolekcji "produkty"
  const productsCollection = db.collection('produkty');

  // Wypełnienie kolekcji "produkty" danymi
  productsCollection.insertMany([
    { nazwa: 'Róża', cena: 15.99, dostępność: 50 },
    { nazwa: 'Tulipan', cena: 10.99, dostępność: 75 },
    { nazwa: 'Lilia', cena: 18.99, dostępność: 30 },
  ], (err, result) => {
    if (err) {
      console.error('Błąd podczas wstawiania danych do kolekcji "produkty":', err);
    } else {
      console.log('Dodano produkty do kolekcji "produkty".');
    }
  });
});
```

NODE.JS

```
// Wypełnienie kolekcji "zamówienia" danymi
ordersCollection.insertOne({
  klient: 'Jan Kowalski',
  produkty: [{ nazwa: 'Róża', ilość: 5 }, { nazwa: 'Lilia', ilość: 2 }],
  data: new Date(),
}, (err, result) => {
  if (err) {
    console.error('Błąd podczas wstawiania danych do kolekcji "zamówienia":', err);
  } else {
    console.log('Dodano zamówienie do kolekcji "zamówienia".');
  }
});

// Zakończenie połączenia z bazą danych
client.close();
});
```



DZIĘKUJĘ ZA UWAGĘ

Tworzenie nierelacyjnej bazy danych w terminalu i podstawowe operacje

Jakub Niemczyk



Tworzenie bazy

```
D:\>USE <database>
```



Wyświetlanie baz

```
D: >show dbs
```



Dodawanie nowych obiektów

```
D:\>db.kolekcja.insert({nazwa: 'obiekt 1'})
```

```
D:\>db.kolekcja.insert({nazwa: 'obiekt 2'})
```



Znajdywanie obiektów

```
D:\>db.kolekcja.find()
```



Modyfikowanie obiektów

```
D:\>db.kolekcja.update({nazwa: obiekt 1}, {nazwa: obiekt 3})
```



Usuwanie obiektów

```
D:\>db.kolekcja.findOneAndDelete()
```



Usuwanie bazy

```
D:\>db.dropDatabase()
```

WYBRANE BIBLIOTEKI DLA MONGODB, ZASTOSOWANIE MONGOOSE

DEFINIWANIE SCHEMATU I MODELU

Mongoose:

```
// Importowanie mongoose i jego schematu
const mongoose = require('mongoose');
const { Schema } = mongoose;

// Definiowanie schematu użytkownika z trzema polami: name, age, email
const userSchema = new Schema({
  name: String, // Typ prosty
  age: Number, // Typ prosty
  email: { type: String, required: true } // Typ z opcją wymagania
});

// Tworzenie modelu na bazie schematu
const User = mongoose.model('User', userSchema);
```

MongoDB:

```
const { MongoClient } = require('mongodb');

async function addUser(name, age, email) {
  const client = new MongoClient(uri);
  try {
    await client.connect();
    const database = client.db('test');
    const users = database.collection('users');

    const newUser = { name, age, email };
    const result = await users.insertOne(newUser);
    console.log(`New user added with id ${result.insertedId}`);
  } finally {
    await client.close();
  }
}
```

DODAWANIE NOWEGO DOKUMENTU

Z Mongoose:

```
// Tworzenie nowej instancji modelu użytkownika
const newUser = new User({
  name: 'Jan Kowalski',
  age: 30,
  email: 'jan@example.com'
});

// Zapisywanie dokumentu do bazy danych
newUser.save()
  .then(doc => console.log(doc))
  .catch(err => console.error(err));
```

Bez Mongoose:

```
// Importowanie Klienta MongoDB
const { MongoClient } = require('mongodb');
const client = new MongoClient(uri);

// Połaczenie z bazą danych i wstawienie dokumentu
client.connect()
  .then(async () => {
    const collection = client.db("test").collection("users");
    await collection.insertOne({ name: 'Jan Kowalski', age: 30, email: 'jan@example.com' });
  })
  .catch(err => console.error(err))
  .finally(() => client.close());
```

WYKONANIE ZAPYTANIA

Z Mongoose:

```
User.find({ age: { $gt: 20 } })
  .then(users => console.log(users))
  .catch(err => console.error(err));
```

Bez Mongoose:

```
// Bezpośrednie wykonanie zapytania do bazy danych
client.connect()
  .then(async () => {
    const collection = client.db("test").collection("users");
    const users = await collection.find({ age: { $gt: 20 } }).toArray();
    console.log(users);
  })
  .catch(err => console.error(err))
  .finally(() => client.close());
```

CREATE (TWORZENIE DANYCH)

Z Mongoose:

```
User.create({ name: 'Jan', age: 30 }, function (err, user) {  
  if (err) console.log(err);  
  else console.log('Utworzono użytkownika:', user);  
});
```

Bez Mongoose:

```
users.insertOne({ name: 'Jan', age: 30 })  
  .then(result => console.log('Utworzono użytkownika'))  
  .catch(err => console.log(err));
```

UPDATE (MODYFIKACJA DANYCH)

Z Mongoose:

```
User.findOneAndUpdate({ name: 'Jan' }, { age: 31 }, function (err, user) {
  if (err) console.log(err);
  else console.log('Zaktualizowano użytkownika:', user);
});
```

Bez Mongoose:

```
users.updateOne({ name: 'Jan' }, { $set: { age: 31 } })
  .then(result => console.log('Zaktualizowano użytkownika'))
  .catch(err => console.log(err));
```

DELETE (USUWANIE DANYCH)

Z Mongoose:

```
User.deleteOne({ name: 'Jan' }, function (err) {  
  if (err) console.log(err);  
  else console.log('Usunięto użytkownika');  
});
```

Bez Mongoose:

```
users.deleteOne({ name: 'Jan' })  
  .then(result => console.log('Usunięto użytkownika'))  
  .catch(err => console.log(err));
```

KRÓTKA CHARAKTERYSTYKA MONK I CAMO

Monk

- ✖ Prostota i intuicyjność
- ✖ Elastyczność
- ✖ Promisy i funkcje zwrotne
- ✖ Łatwa integracja
- ✖ Lekkość

Camo

- ✖ Oparta na Schematach Dokumentów
- ✖ Walidacja i Typowanie
- ✖ Skoncentrowane na Klasach i Obiektach
- ✖ Integracja z nowoczesnym Java Scriptem

ŽRÓDŁA WIEDZY

- ✗ <https://www.scaler.com/topics/mongoose-vs-mongodb/>
- ✗ <https://www.mongodb.com/developer/languages/javascript/getting-started-with-mongodb-and-mongoose/>
- ✗ https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/mongoose
- ✗ <https://stackabuse.com/getting-started-with-camo/>
- ✗ <https://dev.to/franciscomendes10866/an-introduction-to-mongodb-and-node-js-using-monk-3492>

MySQL – instalacja i przedstawienie działania

Szymon Kwaśny, Edukacja Techniczno-Informatyczna, III Rok

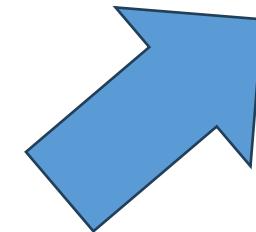
MySQL

- MySQL to otwarty system zarządzania relacyjnymi bazami danych (RDBMS), który został stworzony przez firmę MySQL AB, obecnie rozwijany jest przez firmę Oracle. Jest to jedno z najpopularniejszych narzędzi do przechowywania, zarządzania i pobierania danych w bazach danych.

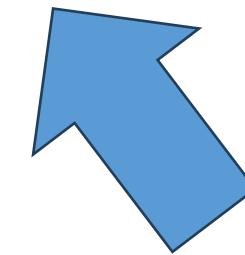


Instalacja MySQL

- Instalacja MySQL



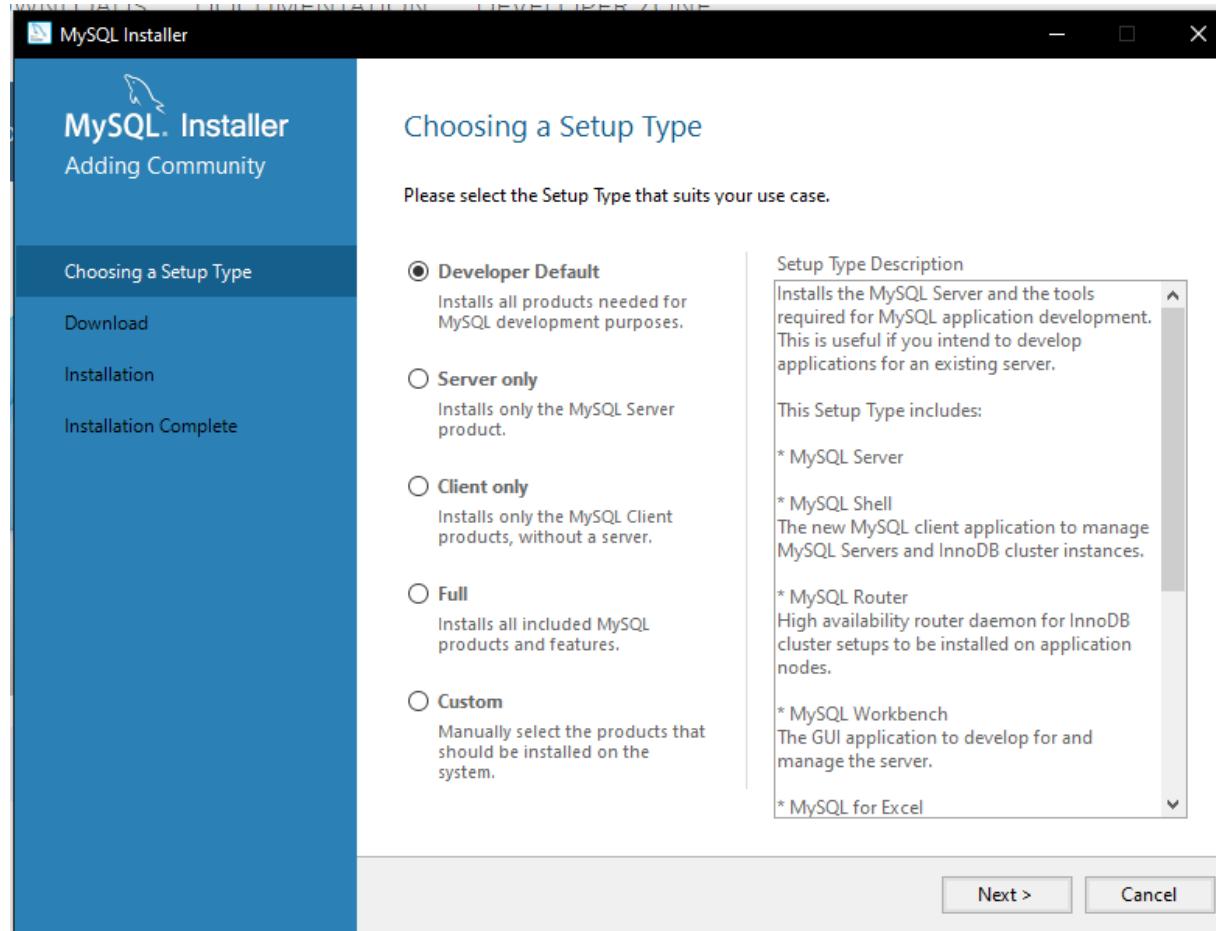
Pobranie pakietu instalacyjnego MySQL



Instalacja gotowych pakietów np. XAMPP



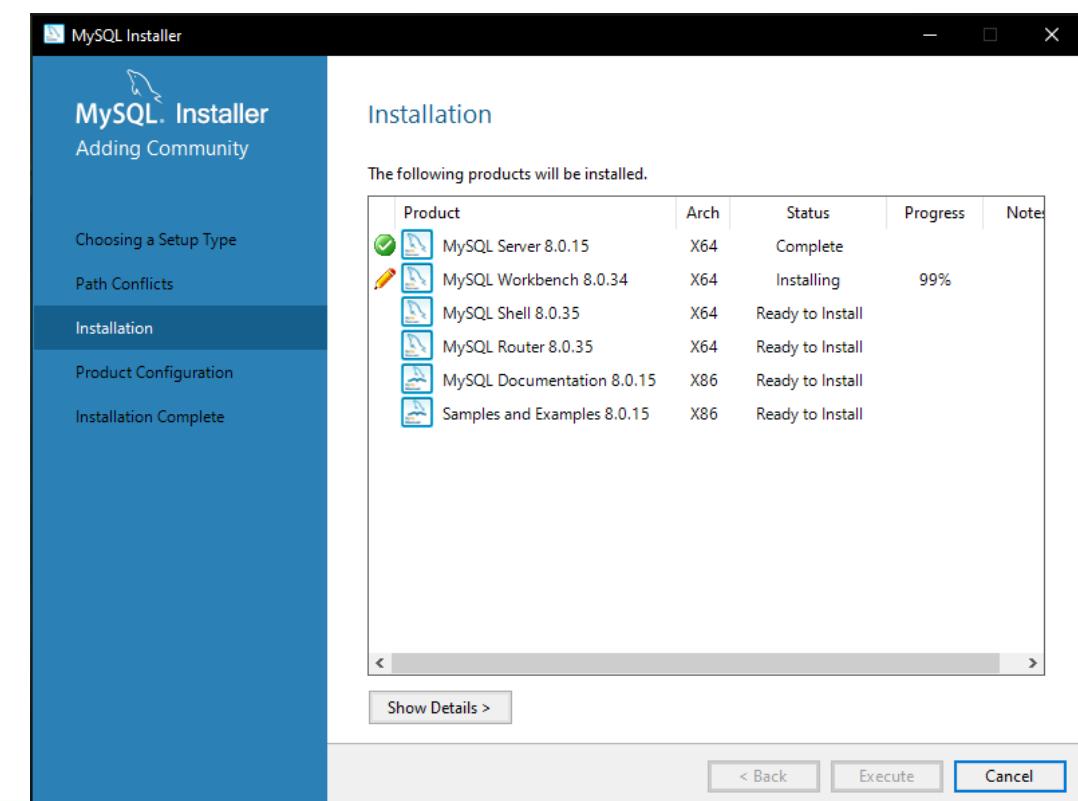
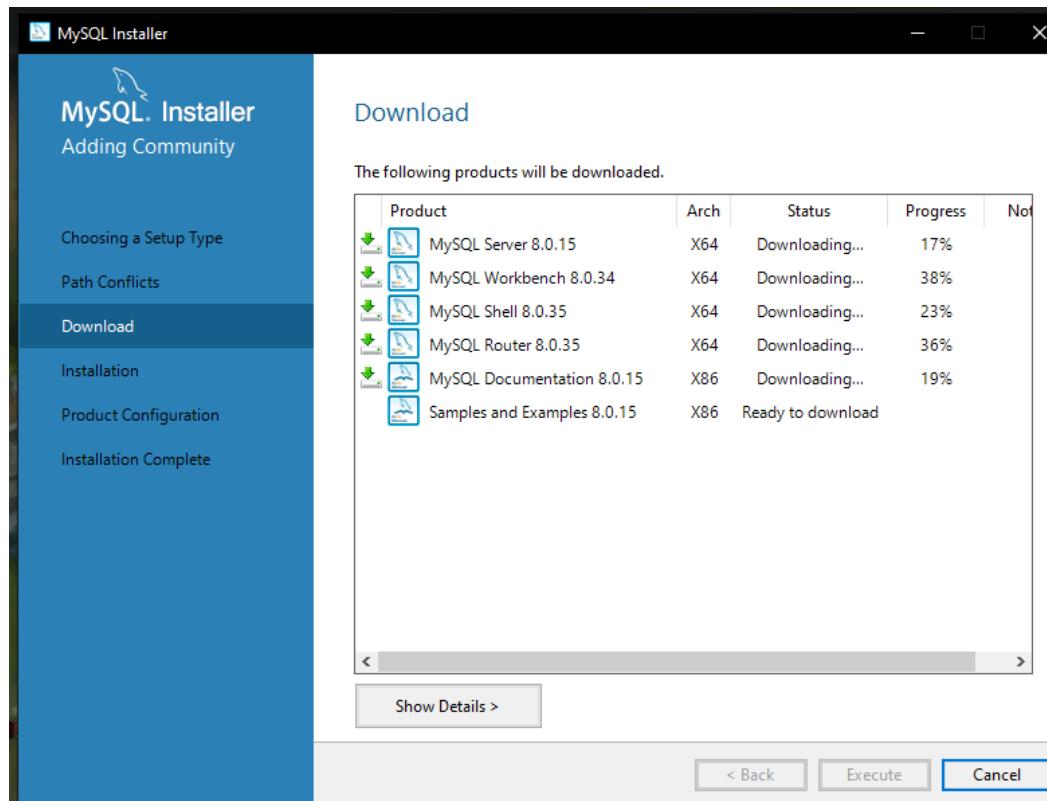
Pakiet instalacyjny MySQL



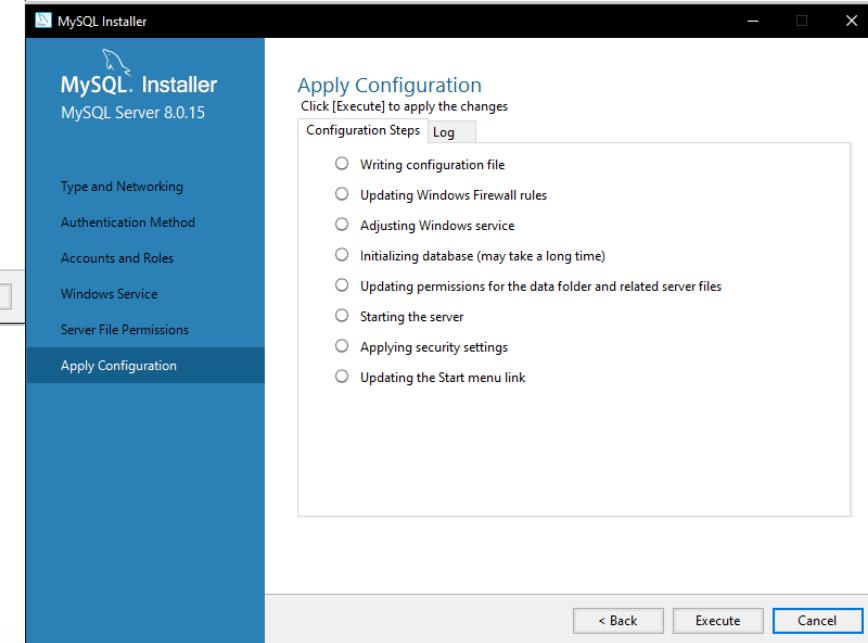
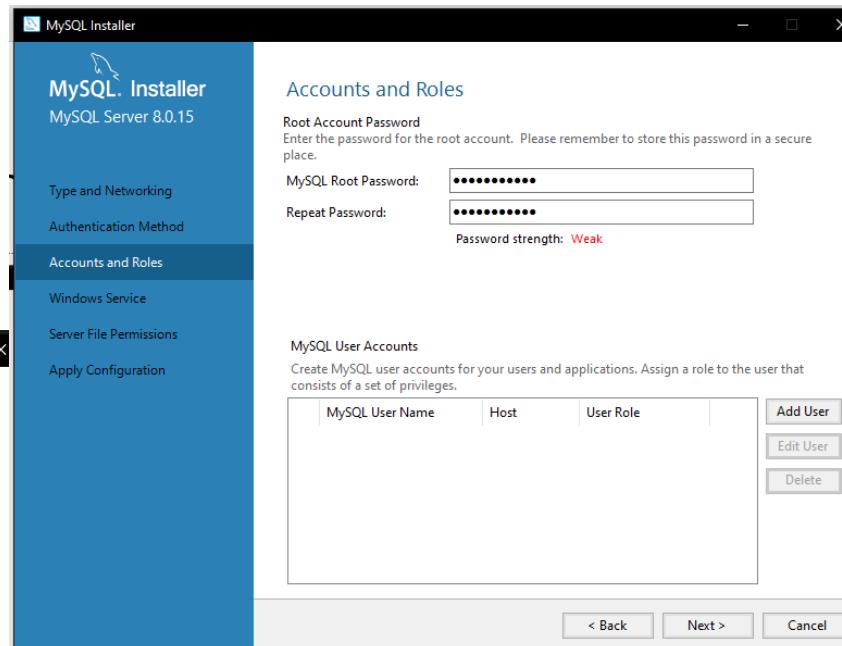
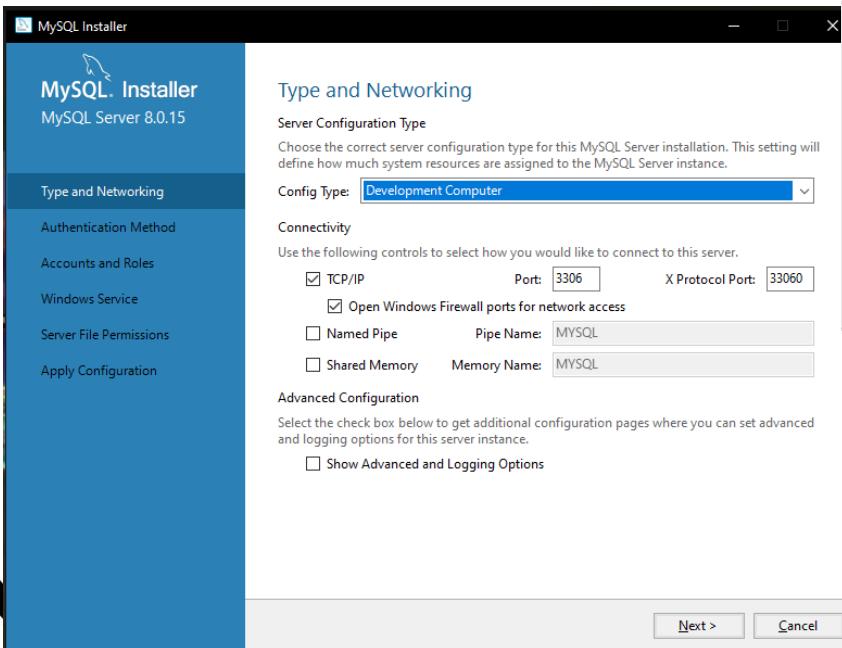
- Instalator przeprowadzi nas przez instalacje oraz pozwoli na skonfigurowanie usługi zgodnie z preferencjami

Pobieranie i instalacja pakietów MySQL

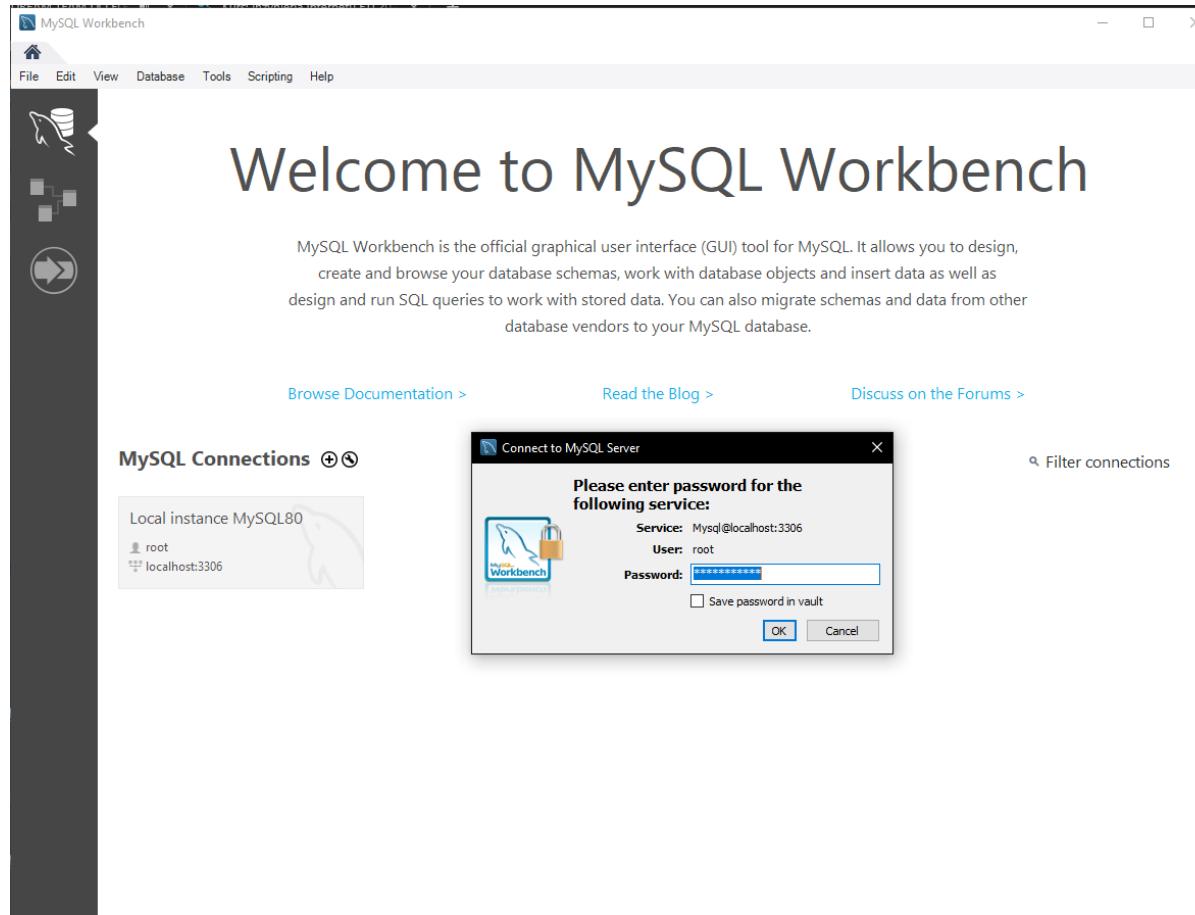
- Instalator pozwala nam wybrać interesujące nas pakiety do pobrania, po czym je zainstaluje



Konfiguracja MySQL



MySQL Workbench



- MySQL Workbench udostępnia nam graficzny interfejs do zarządzania usługą MySQL oraz bazami danych

Korzystanie z MySQL

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Administration - Server Status

Query 1: Administration - Server Status

```
1 -- Tworzenie bazy danych
2 • CREATE DATABASE IF NOT EXISTS SimpleDatabase;
3 • USE SimpleDatabase;
4
5 -- Tworzenie tabeli Users
6 • CREATE TABLE IF NOT EXISTS Users (
7     UserID INT AUTO_INCREMENT PRIMARY KEY,
8     FirstName VARCHAR(50),
9     LastName VARCHAR(50),
10    Email VARCHAR(100) UNIQUE
11 );
12
13 -- Wstawianie danych do tabeli Users
14 • INSERT INTO Users (FirstName, LastName, Email) VALUES
15     ('John', 'Doe', 'john.doe@example.com'),
16     ('Jane', 'Smith', 'jane.smith@example.com'),
17     ('Bob', 'Johnson', 'bob.johnson@example.com');
18
19 -- Tworzenie tabeli Products
20 • CREATE TABLE IF NOT EXISTS Products (
21     ProductID INT AUTO_INCREMENT PRIMARY KEY,
22     ProductName VARCHAR(100),
23     Price DECIMAL(10, 2)
24 );
```

SQLAdditions: Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Output: Action Output

#	Time	Action	Message	Duration / Fetch
1	22:10:16	CREATE DATABASE IF NOT EXISTS SimpleDatabase	1 row(s) affected	0.000 sec
2	22:10:16	USE SimpleDatabase	0 row(s) affected	0.000 sec
3	22:10:16	CREATE TABLE IF NOT EXISTS Users (UserID INT AUTO_INCREMENT PRIMARY KEY, FirstName VARCHAR(50), LastName VARCHAR(50), Email VARCHAR(100) UNIQUE)	0 row(s) affected	0.031 sec
4	22:10:16	INSERT INTO Users (FirstName, LastName, Email) VALUES ('John', 'Doe', 'john.doe@example.com')	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.000 sec
5	22:10:16	CREATE TABLE IF NOT EXISTS Products (ProductID INT AUTO_INCREMENT PRIMARY KEY, ProductName VARCHAR(100), Price DECIMAL(10, 2))	0 row(s) affected	0.015 sec
6	22:10:16	INSERT INTO Products (ProductName, Price) VALUES ('Laptop', 1200.00)	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.000 sec
7	22:10:16	CREATE TABLE IF NOT EXISTS Orders (OrderID INT AUTO_INCREMENT PRIMARY KEY, UserID INT, ProductID INT, OrderDate DATE, FOREIGN KEY (UserID) REFERENCES Users(UserID), FOREIGN KEY (ProductID) REFERENCES Products(ProductID))	0 row(s) affected	0.032 sec
8	22:10:16	INSERT INTO Orders (UserID, ProductID, OrderDate) VALUES (1, 1, '2023-01-01')	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.000 sec

Object Info Session

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Administration - Server Status

Query 1: Administration - Server Status

```
24 );
25
26 -- Wstawianie danych do tabeli Products
27 • INSERT INTO Products (ProductName, Price) VALUES
28     ('Laptop', 1200.00),
29     ('Smartphone', 599.99),
30     ('Headphones', 79.99);
31
32 -- Tworzenie tabeli Orders
33 • CREATE TABLE IF NOT EXISTS Orders (
34     OrderID INT AUTO_INCREMENT PRIMARY KEY,
35     UserID INT,
36     ProductID INT,
37     OrderDate DATE,
38     FOREIGN KEY (UserID) REFERENCES Users(UserID),
39     FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
40 );
41
42 -- Wstawianie danych do tabeli Orders
43 • INSERT INTO Orders (UserID, ProductID, OrderDate) VALUES
```

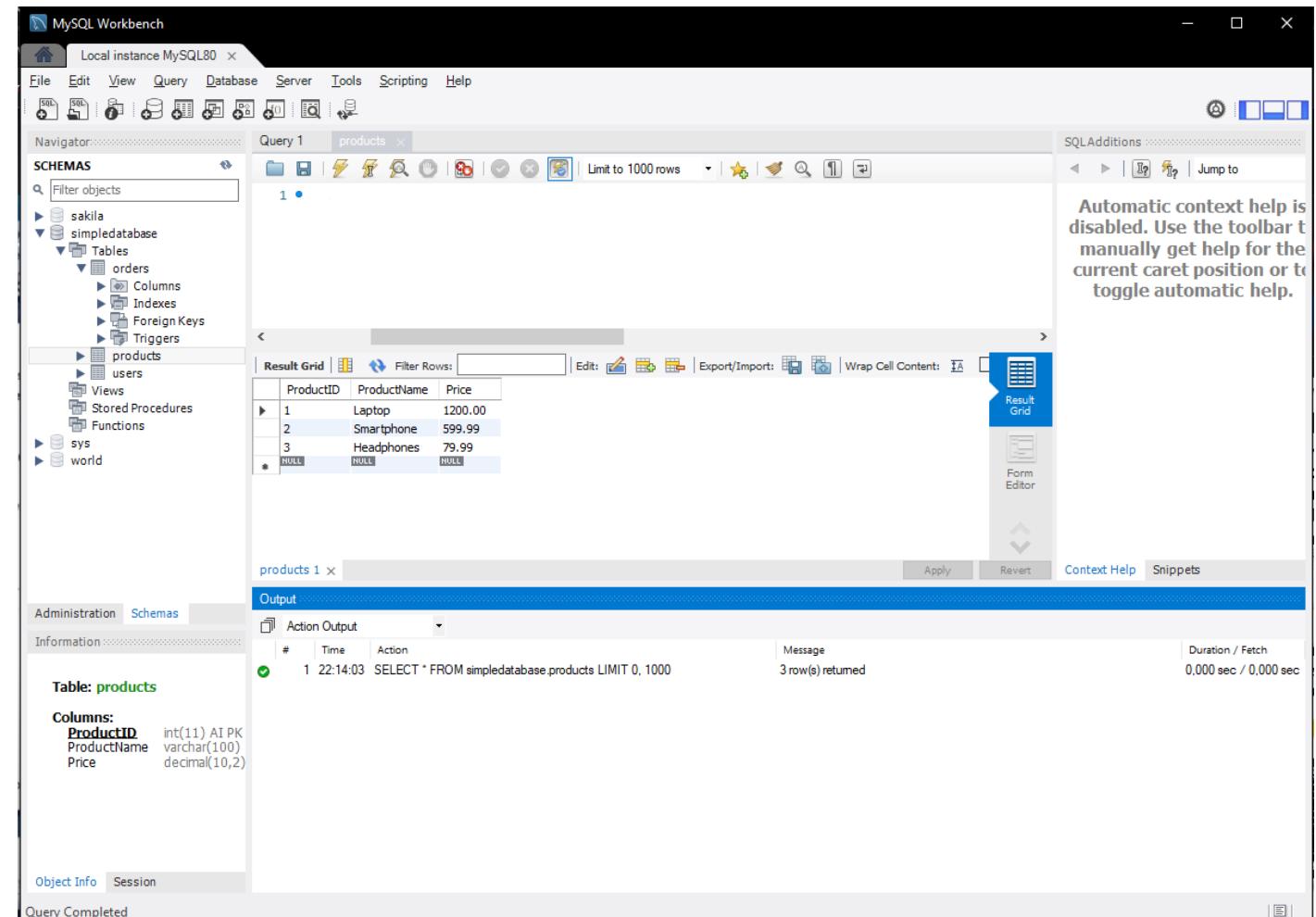
Output: Action Output

#	Time	Action	Message	Duration / Fetch
1	22:10:16	CREATE DATABASE IF NOT EXISTS SimpleDatabase	1 row(s) affected	0.000 sec
2	22:10:16	USE SimpleDatabase	0 row(s) affected	0.000 sec
3	22:10:16	CREATE TABLE IF NOT EXISTS Users (UserID INT AUTO_INCREMENT PRIMARY KEY, FirstName VARCHAR(50), LastName VARCHAR(50), Email VARCHAR(100) UNIQUE)	0 row(s) affected	0.031 sec
4	22:10:16	INSERT INTO Users (FirstName, LastName, Email) VALUES ('John', 'Doe', 'john.doe@example.com')	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.000 sec
5	22:10:16	CREATE TABLE IF NOT EXISTS Products (ProductID INT AUTO_INCREMENT PRIMARY KEY, ProductName VARCHAR(100), Price DECIMAL(10, 2))	0 row(s) affected	0.015 sec
6	22:10:16	INSERT INTO Products (ProductName, Price) VALUES ('Laptop', 1200.00)	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.000 sec
7	22:10:16	CREATE TABLE IF NOT EXISTS Orders (OrderID INT AUTO_INCREMENT PRIMARY KEY, UserID INT, ProductID INT, OrderDate DATE, FOREIGN KEY (UserID) REFERENCES Users(UserID), FOREIGN KEY (ProductID) REFERENCES Products(ProductID))	0 row(s) affected	0.032 sec
8	22:10:16	INSERT INTO Orders (UserID, ProductID, OrderDate) VALUES (1, 1, '2023-01-01')	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.000 sec

Object Info Session

Podgląd projektu

- MySQL Workbench umożliwia nam zobaczenia poprzez interfejs projekt naszej bazy danych, jej tabele, klucze obce itp.



Instalacja XAMPP

- Alternatywą jest pobranie pakietu XAMPP, który zawiera usługę MySQL. Instalator sam skonfiguruje najważniejsze ustawienia

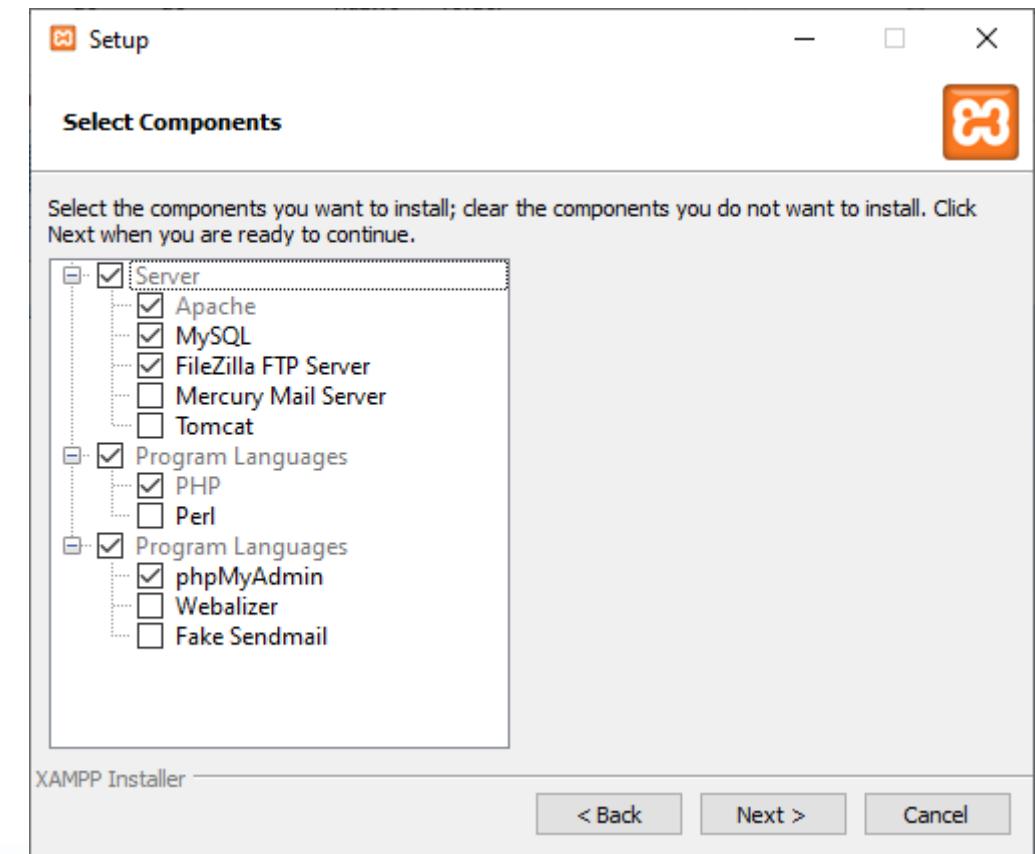


XAMPP dla Windows 8.0.28, 8.1.17 & 8.2.4

Wersja	Suma kontrolna	Rozmiar
8.0.28 / PHP 8.0.28	Co zawiera? md5 sha1	Pobierz (64 bit) 144 Mb
8.1.17 / PHP 8.1.17	Co zawiera? md5 sha1	Pobierz (64 bit) 148 Mb
8.2.4 / PHP 8.2.4	Co zawiera? md5 sha1	Pobierz (64 bit) 149 Mb

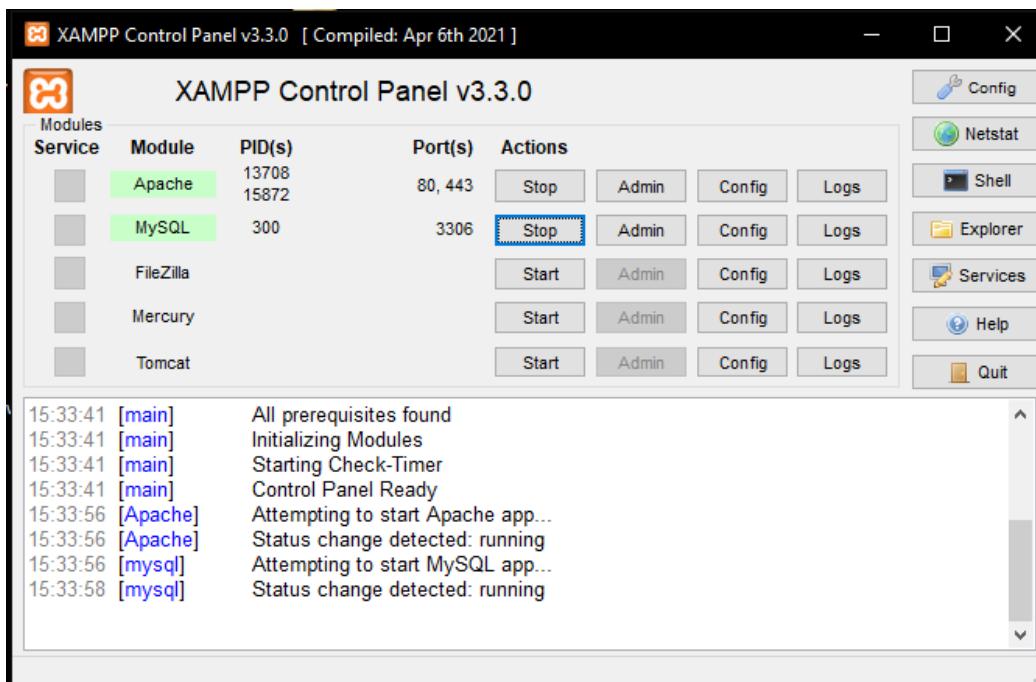
[Wymagania komputerowe](#) [Więcej opcji pobierania »](#)

Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms [here](#).



Korzystanie z XAMPP

- Po zainstalowaniu XAMPPa należy uruchomić usługę Apache oraz MySQL, następnie przy pomocy opcji shell otworzyć linie poleceń. Aby uruchomić MySQL należy wpisać polecania cd mysql/bin -> mysql –u root



The screenshot shows a terminal window titled "XAMPP for Windows - mysql -u root". It displays the MySQL command-line interface. The user has run the command "cd mysql/bin" and then "mysql -u root". The MySQL prompt "MariaDB [(none)]>" is visible at the bottom.

```
Setting environment for using XAMPP for Windows.
Kwaśny@DESKTOP-2M19CQG d:\Studies\app\xampp
# cd mysql/bin

Kwaśny@DESKTOP-2M19CQG d:\Studies\app\xampp\mysql\bin
# mysql -u root
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 11
Server version: 10.4.27-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Interfejs graficzny

- XAMPP oferuje również graficzny interfejs, który umożliwia tworzenie oraz zarządzanie bazą danych. W oknie przeglądarki należy wpisać adres localhost a następnie wybrać opcję phpMyAdmin



Welcome to XAMPP for Windows 8.2.0

You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the FAQs section or check the HOW-TO Guides for getting started with PHP applications.

XAMPP is meant only for development purposes. It has certain configuration settings that make it easy to develop locally but that are insecure if you want to have your installation accessible to others.

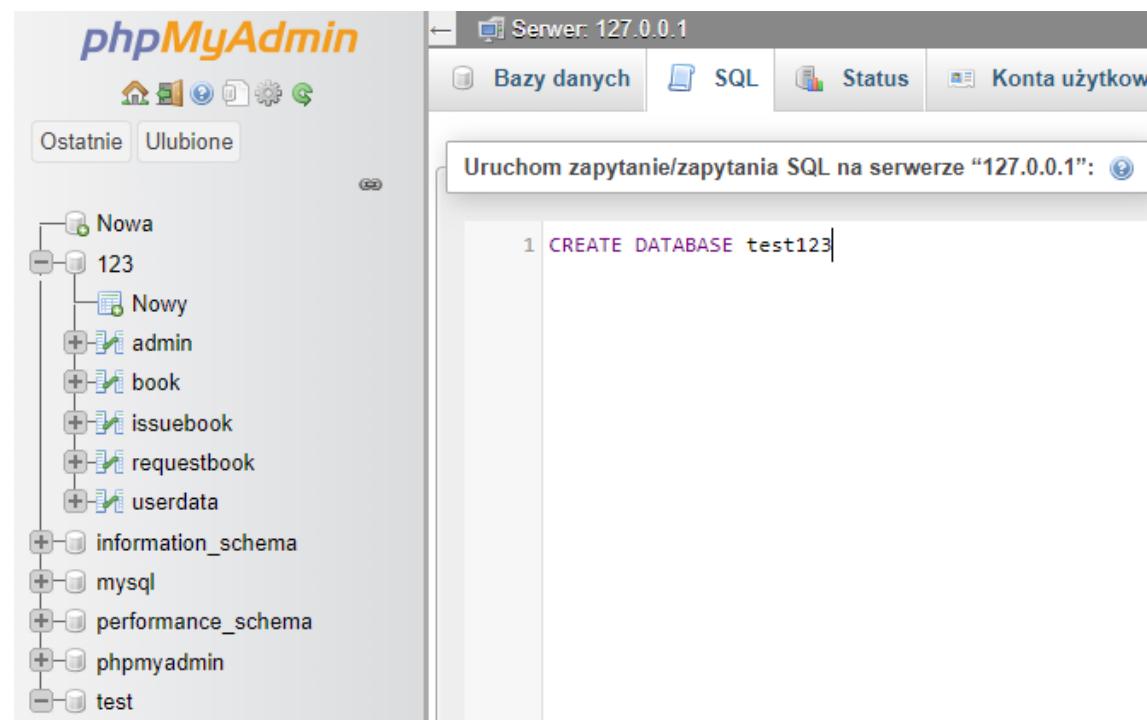
Start the XAMPP Control Panel to check the server status.

Community

XAMPP has been around for more than 10 years – there is a huge community behind it. You can get involved by joining our Forums, liking us on Facebook, or following our exploits on Twitter.

The screenshot shows the phpMyAdmin interface running on a local server (127.0.0.1). The main dashboard includes sections for general settings, appearance, and server information. On the left, a sidebar lists databases: Nova, 123, information_schema, mysql, performance_schema, phpmyadmin, and test. A message at the bottom indicates a new version of phpMyAdmin is available for update. The URL in the address bar is localhost/phpmyadmin/index.php?route=/database/operations&db=123.

Utworzenie bazy danych



The screenshot shows the phpMyAdmin interface for a MySQL server at 127.0.0.1. The left sidebar lists existing databases: Nowa, 123, Nowy, admin, book, issuebook, requestbook, userdata, information_schema, mysql, performance_schema, phpmyadmin, and test. The main area has tabs for 'Bazy danych' (selected), 'SQL', 'Status', and 'Konta użytkowni'. A SQL query window displays the command: 'CREATE DATABASE test123'. The output window shows the results of the query:

```
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 56
Server version: 10.4.27-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE test123
                  -> ;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]>
```

Tworzenie tabeli

phpMyAdmin

Serwer: 127.0.0.1 » Baza danych: 123

Struktura SQL Szukaj Zapytanie Eksport

Nie ma tabel ulubionych.

Ostatnie Ulubione

Nowa 123 Nowy admin book issuebook requestbook userdata

information_schema mysql performance_schema phpmyadmin test test123

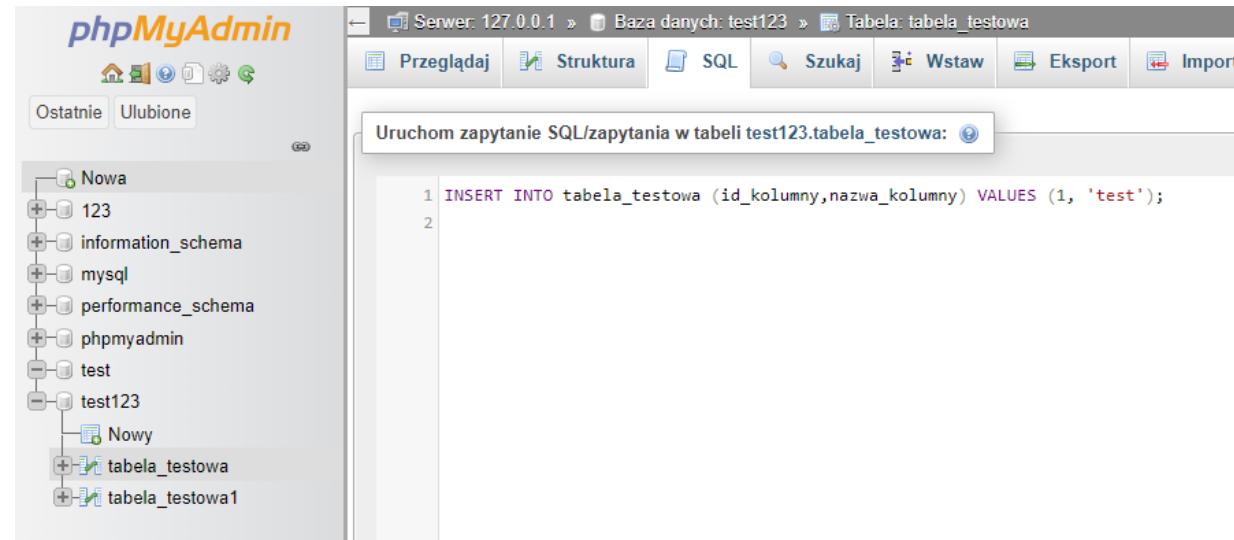
Wykonanie zapytania/zapytań SQL do bazy danych 123:

```
1 CREATE TABLE tabela_testowa (
2     id_kolumny INT(3),
3     nazwa_kolumny VARCHAR(25)
4 );
```

```
MariaDB [(none)]> Use test123
Database changed
MariaDB [test123]> CREATE TABLE tabela_testowa1 (
    ->     id_kolumny INT(3),
    ->     nazwa_kolumny VARCHAR(25)
    -> );
Query OK, 0 rows affected (0.502 sec)

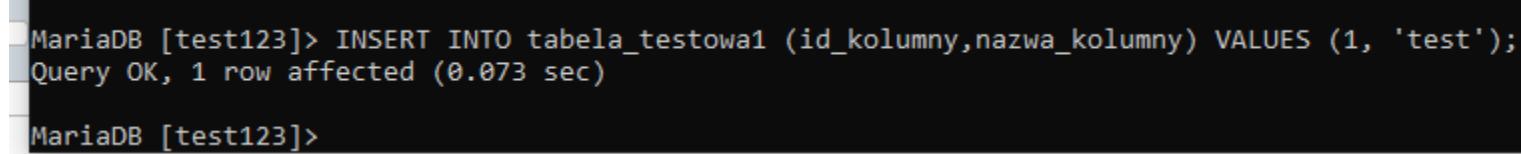
MariaDB [test123]>
```

Wstawianie danych



The screenshot shows the phpMyAdmin interface. The left sidebar displays a tree view of databases and tables. The current database selected is 'test123'. Inside 'test123', there are two tables: 'Nowy' and 'tabela_testowa'. The 'tabela_testowa' table is currently selected. The top navigation bar shows the server as 'Serwer: 127.0.0.1', the database as 'Baza danych: test123', and the table as 'Tabela: tabela_testowa'. Below the navigation bar, there are several tabs: Przeglądaj, Struktura, SQL, Szukaj, Wstaw, Eksport, and Import. The 'SQL' tab is active. In the main area, there is a text input field labeled 'Uruchom zapytanie SQL/zapytania w tabeli test123.tabela_testowa:' containing the following SQL code:

```
1 INSERT INTO tabela_testowa (id_kolumny,nazwa_kolumny) VALUES (1, 'test');
```



```
MariaDB [test123]> INSERT INTO tabela_testowa1 (id_kolumny,nazwa_kolumny) VALUES (1, 'test');
Query OK, 1 row affected (0.073 sec)

MariaDB [test123]>
```

Wyświetlanie danych

Pokaż okno zapytań

⚠️ Bieżący wybór nie zawiera unikalnej kolumny. Opcje edycji siatki, pola wyboru, edycji, kopowania i usuwania są niedostępne.

✓ Pokazano wiersze 0 - 0 (1 ogółem, Wykonanie zapytania trwało 0,0001 sekund(y).)

```
SELECT * FROM tabela_testowa1;
```

Profilowanie [Edytuj w linii] [Edytuj] [Wyjaśnij SQL] [Utwórz kod PHP] [Odśwież]

Pokaż wszystko | Liczba wierszy: 25 | Filtrowanie wierszy: Przeszukaj tę tabelę

Extra options

id_kolumny	nazwa_kolumny
1	test

Pokaż wszystko | Liczba wierszy: 25 | Filtrowanie wierszy: Przeszukaj tę tabelę

```
MariaDB [test123]> SELECT * FROM `tabela_testowa1`  
-> ;  
+-----+-----+  
| id_kolumny | nazwa_kolumny |  
+-----+-----+  
| 1 | test |  
+-----+-----+  
1 row in set (0.000 sec)  
  
MariaDB [test123]>
```

Importowanie bazy danych

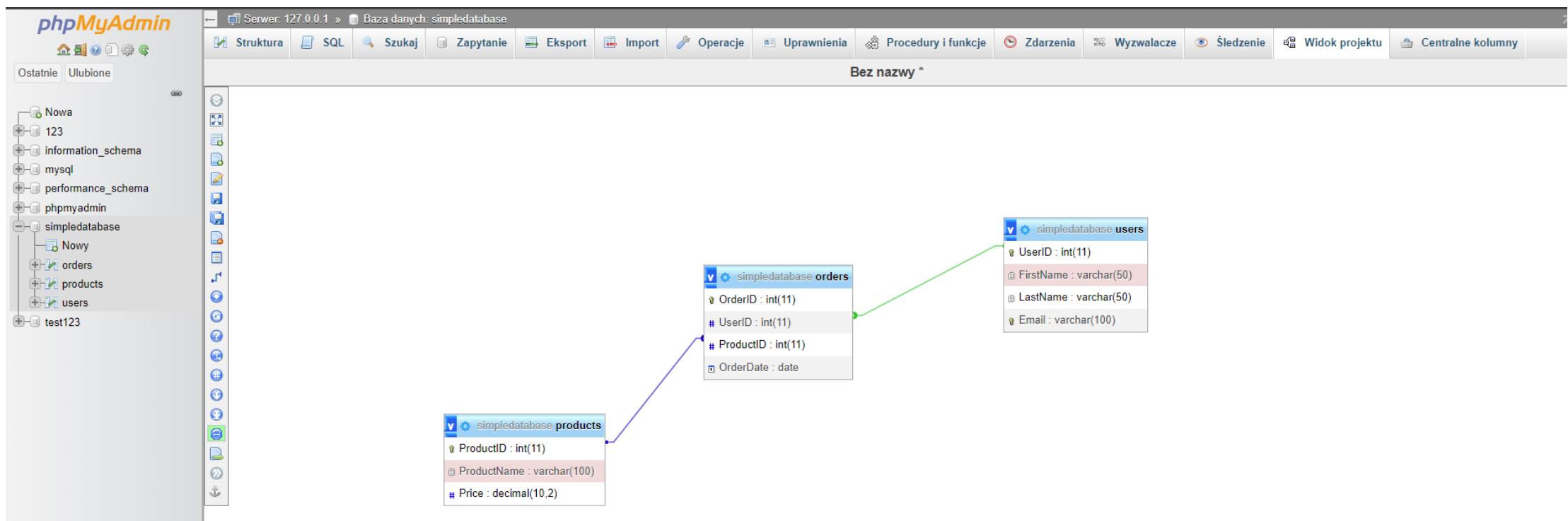
- MySQL umożliwia nam zimportowanie wcześniej utworzonej bazy danych.



The screenshot shows the phpMyAdmin interface for importing databases. The top navigation bar includes tabs for 'Bazy danych', 'SQL', 'Status', 'Konta użytkowników', 'Eksport', 'Import' (which is currently selected), 'Ustawienia', 'Replikacja', 'Zmienne', 'Kodowania znaków', 'Mechanizmy', and 'Wtyczki'. The left sidebar displays a tree view of databases, including 'Nowa', '123', 'Nowy', 'admin', 'book', 'issuebook', 'requestbook', 'tabela_testowa', 'userdata', 'information_schema', 'mysql', 'performance_schema', 'phpmyadmin', and 'test123'. The main content area is titled 'Importowanie do bieżącego serwera' (Import to current server). It features a file input field labeled 'Plik do importu:' (File to import:) with the placeholder text 'Plik może być skompresowany (gzip, bzip2) bądź nie. Nazwa skompresowanego pliku musi kończyć się na .[format].[compression]. Przykład: .sql.zip'. Below this is a search field 'Wyszukaj w komputerze: (Maksymalny rozmiar: 40MB)' (Search on computer: (Maximum size: 40MB)). A file selection button 'Wybierz plik' (Select file) shows 'database.sql'. There is also a note 'Możesz także przeciągnąć i upuścić plik na dowolnej stronie.' (You can also drag and drop the file anywhere on the page.). At the bottom, there is a dropdown menu for 'Kodowanie znaków pliku:' (Character encoding of the file:) set to 'utf-8'.

Widok projektu w phpMyAdmin

- Mamy również możliwość zobaczenie, jak wygląda nasza baza danych, opcja widok projektu wyświetla stworzone tabele oraz pokazuje utworzone między nimi relacje



Bibliografia

- <https://www.w3schools.com/sql/> [14.11.2023]
- <https://www.mysql.com/> [14.11.2023]
- https://informatyka.2ap.pl/ftp/technik_inf/kurs_mysql.pdf [14.11.2023]
- <https://strefakodera.pl/bazy-danych/sql/tworzenie-usuwanie-i-modyfikacja-tabel-w-sql> [14.11.2023]
- <https://zaprogramujzycie.pl/jak-zainstalowac-xampp/> [14.11.2023]

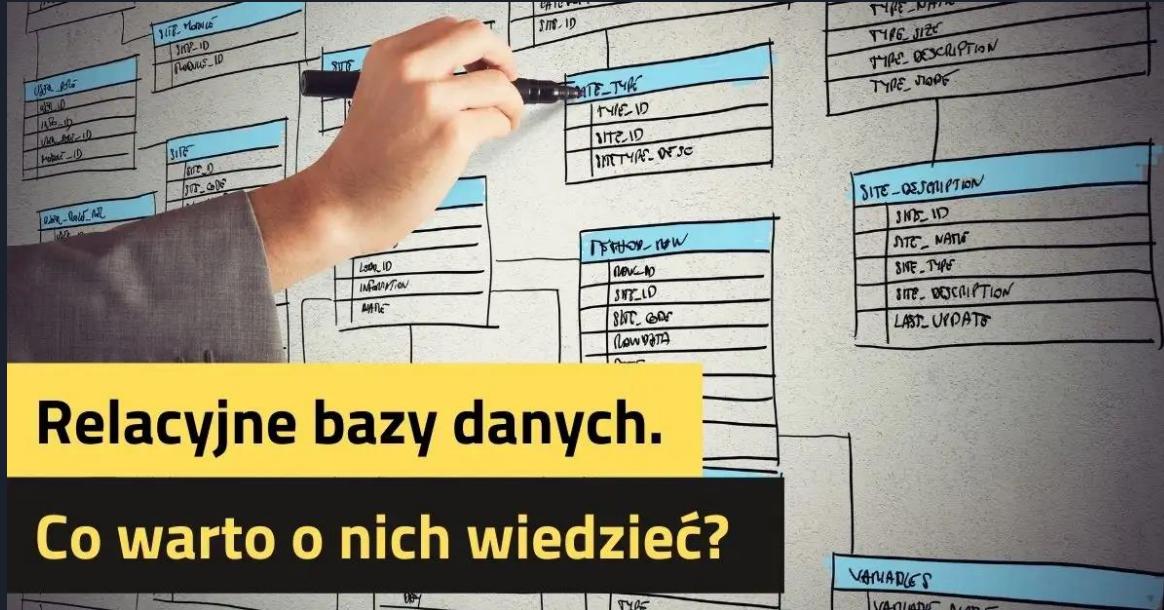


Stworzenie relacyjnej bazy danych z linii poleceń - podstawowe operacje na bazie

Kacper Słapak III rok ETI

Czym jest relacyjna baza danych?

Relacyjna baza danych (RDBMS) to rodzaj systemu zarządzania bazą danych (DBMS), który opiera się na modelu relacyjnym danych. Model relacyjny został stworzony przez Edgara Codd'a w 1970 roku, a później został rozwinięty i udoskonalony przez różnych badaczy i praktyków.





Kluczowe cechy relacyjnych baz danych

Tabela:

- Dane są przechowywane w postaci tabel, które składają się z wierszy i kolumn.
- Każda tabela reprezentuje pewien rodzaj obiektu lub koncepcji, a kolumny tabeli odpowiadają atrybutom tego obiektu.

Klucz główny:

- Każda tabela ma jednoznacznie zidentyfikowany klucz główny.
- Klucz główny służy do unikatowej identyfikacji każdego rekordu w tabeli.

Relacje:

- Tabele w relacyjnej bazie danych są ze sobą powiązane relacjami.
- Relacje między tabelami są nawiązywane za pomocą kluczy obcych.



Kluczowe cechy relacyjnych baz danych cd

Integralność referencyjna:

- Systemy relacyjne zapewniają integralność referencyjną, co oznacza, że wartości w kluczach obcych muszą mieć swoje odpowiedniki w kluczach głównych innych tabel.

Język zapytań SQL:

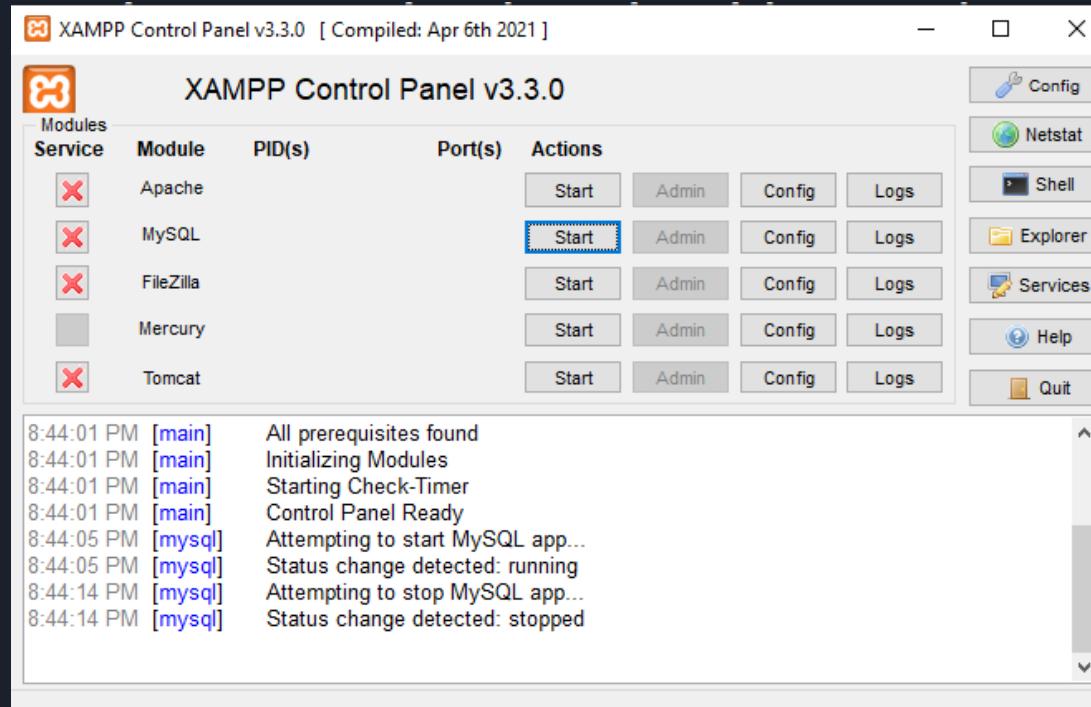
- Do komunikacji z relacyjną bazą danych używa się języka zapytań SQL (Structured Query Language).
- SQL umożliwia tworzenie, modyfikowanie, wyszukiwanie i usuwanie danych z bazy.

Normalizacja:

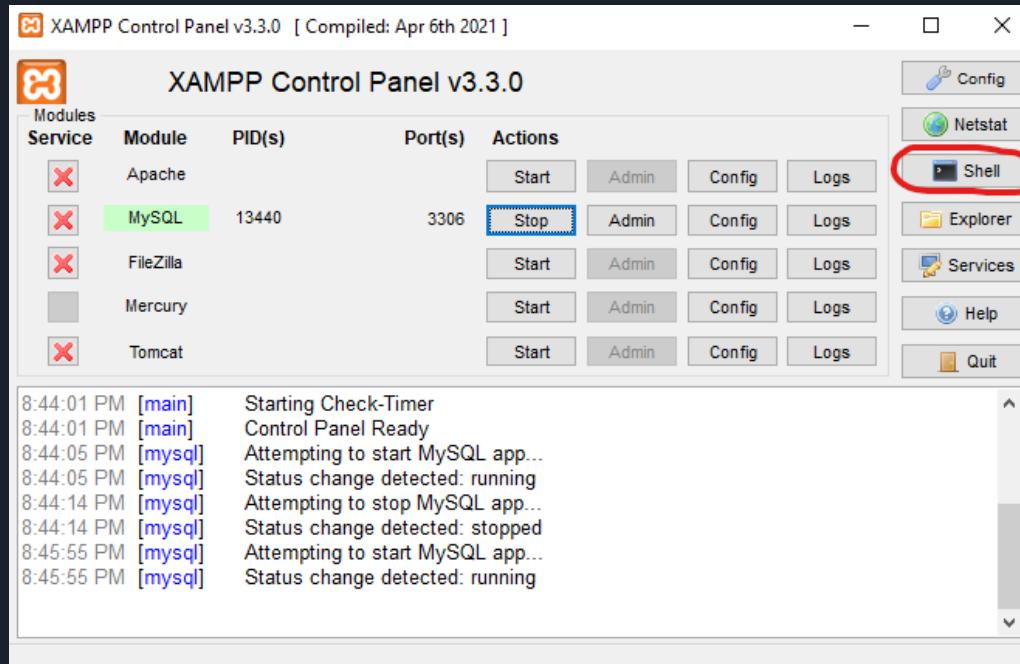
- Relacyjne bazy danych często są zaprojektowane zgodnie z zasadami normalizacji, co oznacza, że dane są podzielone na logiczne jednostki w celu uniknięcia redundancji i utrzymania integralności danych.

Utworzenie prostej relacyjnej bazy danych

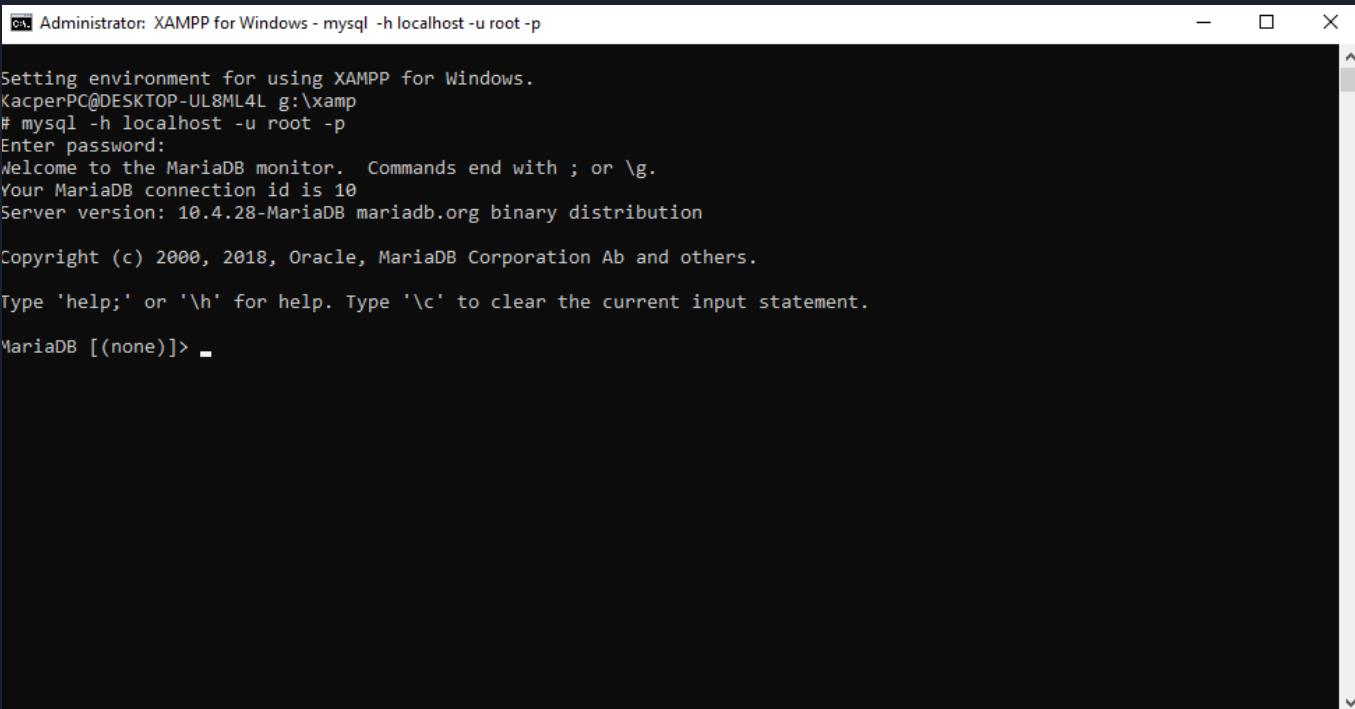
1. Używając np programu XAMPP uruchamiamy moduł MySQL



2. Następnie uruchamiamy wiersz poleceń



3. Używamy polecenia: *mysql -h localhost -u root -p*



The screenshot shows a Windows Command Prompt window titled "Administrator: XAMPP for Windows - mysql -h localhost -u root -p". The window displays the MySQL command-line interface. The output includes the environment setup message, the user's connection information (KacperPC@DESKTOP-UL8ML4L), the MariaDB monitor welcome message, the server version (10.4.28-MariaDB mariadb.org binary distribution), copyright information, help instructions, and the prompt "MariaDB [(none)]>".

```
Administrator: XAMPP for Windows - mysql -h localhost -u root -p

Setting environment for using XAMPP for Windows.
KacperPC@DESKTOP-UL8ML4L g:\xampp
# mysql -h localhost -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 10
Server version: 10.4.28-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

4. Do stworzenia naszej bazy używamy polecenia: ***create database "DowolnaNazwa";***

Możemy również skorzystać z komendy: ***show databases;*** aby sprawdzić czy baza o podanej przez nas nazwie została poprawnie utworzona



```
Administrator: XAMPP for Windows - mysql -h localhost -u root -p

Setting environment for using XAMPP for Windows.
KacperPC@DESKTOP-UL8ML4L g:\xamp
# mysql -h localhost -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 10
Server version: 10.4.28-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database Przyklad;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> show databases;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| phpmyadmin     |
| przyklad       |
| test           |
+-----+
6 rows in set (0.001 sec)

MariaDB [(none)]> -
```

5. Aby móc zacząć operować na naszej nowo utworzonej bazie korzystamy z polecenia: use "DowolnaNazwa";

Otrzymamy komunikat o zmianie aktualnie używanej bazy danych.

```
MariaDB [(none)]> use przyklad;
Database changed
MariaDB [przyklad]> ■
```

6. Tworzymy naszą pierwszą tabelę, w tym celu używamy: *create table “DowolnaNazwa”* z odpowiednimi parametrami jak na przykładzie

Ponownie możemy sprawdzić poprawność wpisanej komendy wywołując utworzoną tablicę

```
MariaDB [(none)]> use przyklad;
Database changed
MariaDB [przyklad]> create table uczniowie(
    -> id int(11) not null AUTO_INCREMENT PRIMARY KEY,
    -> imie VARCHAR(100) not null,
    -> nazwisko VARCHAR(100) not null,
    -> klasa VARCHAR(100) not null
    -> );
Query OK, 0 rows affected (0.209 sec)
```

```
MariaDB [przyklad]> desc uczniowie
    -> ;
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key  | Default | Extra           |
+-----+-----+-----+-----+-----+
| id   | int(11)   | NO   | PRI  | NULL    | auto_increment |
| imie | varchar(100)| NO   |       | NULL    |                 |
| nazwisko | varchar(100)| NO   |       | NULL    |                 |
| klasa | varchar(100)| NO   |       | NULL    |                 |
+-----+-----+-----+-----+-----+
4 rows in set (0.047 sec)
```

```
MariaDB [przyklad]>
```

7. Tworzymy naszą drugą tabelę, w tym celu używamy również : *create table “DowolnaNazwa”* z odpowiednimi parametrami jak na przykładzie

Tym razem ustalamy dodatkowy parametr będący kluczem obcym odnoszącym się do pierwszej tabeli.

```
MariaDB [przyklad]> CREATE TABLE Oceny(
    -> OcenaID int primary key auto_increment,
    -> UczenID int,\n    -> Przedmiot VARCHAR(50),
    -> Ocena int,
    -> foreign key (UczenID) references uczniowie(id)
    -> );
Query OK, 0 rows affected (0.211 sec)

MariaDB [przyklad]> desc Oceny;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra        |
+-----+-----+-----+-----+-----+-----+
| OcenaID | int(11) | NO   | PRI  | NULL    | auto_increment |
| UczenID | int(11) | YES  | MUL  | NULL    |                |
| Przedmiot | varchar(50) | YES  |       | NULL    |                |
| Ocena | int(11) | YES  |       | NULL    |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.013 sec)

MariaDB [przyklad]> -
```

Podstawowe operacje na bazie danych

1. Dodanie danych do tabeli Uczniowie

```
MariaDB [przyklad]> insert into uczniowie (id, imie, nazwisko, klasa) values
    ->      (1, 'Jan', 'Kowalski', '3A');
Query OK, 1 row affected (0.236 sec)

MariaDB [przyklad]> insert into uczniowie (id, imie, nazwisko, klasa) values
    ->      (2, 'Anna', 'Nowak', '2B');
Query OK, 1 row affected (0.050 sec)

MariaDB [przyklad]> select * from uczniowie;
+----+-----+-----+----+
| id | imie | nazwisko | klasa |
+----+-----+-----+----+
|  1 | Jan  | Kowalski | 3A   |
|  2 | Anna | Nowak   | 2B   |
+----+-----+-----+----+
2 rows in set (0.000 sec)
```

Podstawowe operacje na bazie danych

2. Dodanie danych do tabeli Oceny, odnosząc się do uczniów za pomocą klucza obcego

```
MariaDB [przyklad]> INSERT INTO Oceny (OcenaID, UczenID, Przedmiot, Ocena) VALUES
    ->      (101, 1, 'Matematyka', 4),
    ->      (102, 2, 'Historia', 5);
Query OK, 2 rows affected (0.046 sec)
Records: 2  Duplicates: 0  Warnings: 0

MariaDB [przyklad]> select * from Oceny;
+-----+-----+-----+-----+
| OcenaID | UczenID | Przedmiot | Ocena |
+-----+-----+-----+-----+
|     101 |        1 | Matematyka |      4 |
|     102 |        2 | Historia   |      5 |
+-----+-----+-----+-----+
2 rows in set (0.000 sec)
```

Podstawowe operacje na bazie danych

3. Wyświetlanie danych z obu tabel za pomocą złączenia JOIN

```
MariaDB [przyklad]> SELECT uczniowie.imie, uczniowie.nazwisko, Oceny.Predmiot, Oceny.Ocena
-> FROM uczniowie
-> JOIN Oceny ON uczniowie.id = Oceny.UczenID;
+-----+-----+-----+
| imie | nazwisko | Predmiot | Ocena |
+-----+-----+-----+
| Jan  | Kowalski | Matematyka |      4 |
| Anna | Nowak   | Historia  |      5 |
+-----+-----+-----+
2 rows in set (0.000 sec)
```

Podstawowe operacje na bazie danych

4. Aktualizacja danych w tabeli

```
MariaDB [przyklad]> select * from uczniowie  
-> ;  
+---+-----+-----+-----+  
| id | imie | nazwisko | klasa |  
+---+-----+-----+-----+  
| 1 | Jan | Kowalski | 3A |  
| 2 | Anna | Nowak | 2B |  
+---+-----+-----+  
2 rows in set (0.000 sec)  
  
MariaDB [przyklad]> update uczniowie set klasa = '3B' where id = 1;  
Query OK, 1 row affected (0.048 sec)  
Rows matched: 1  Changed: 1  Warnings: 0  
  
MariaDB [przyklad]> select * from uczniowie ;  
+---+-----+-----+-----+  
| id | imie | nazwisko | klasa |  
+---+-----+-----+-----+  
| 1 | Jan | Kowalski | 3B |  
| 2 | Anna | Nowak | 2B |  
+---+-----+-----+-----+  
2 rows in set (0.000 sec)
```

Podstawowe operacje na bazie danych

5. Usunięcie danych z tabeli oceny

```
MariaDB [przyklad]> select * from Oceny;
+-----+-----+-----+
| OcenaID | UczenID | Przedmiot | Ocena |
+-----+-----+-----+
|      101 |        1 | Matematyka |      4 |
|      102 |        2 | Historia   |      5 |
+-----+-----+-----+
2 rows in set (0.000 sec)

MariaDB [przyklad]> DELETE FROM Oceny WHERE OcenaID = 102;
Query OK, 1 row affected (0.047 sec)

MariaDB [przyklad]> select * from Oceny;
+-----+-----+-----+
| OcenaID | UczenID | Przedmiot | Ocena |
+-----+-----+-----+
|      101 |        1 | Matematyka |      4 |
+-----+-----+-----+
1 row in set (0.000 sec)
```

Połączenie aplikacji z nierelacyjną bazą danych.

Radosław Żyła

MongoDB Atlas

MongoDB Atlas to usługa chmurowa oferująca zarządzanie bazą danych MongoDB w chmurze. Zapewnia skalowalność, wysoką dostępność i bezpieczeństwo danych.



```
PS C:\Users\radek> npm install mongodb
```

```
PS C:\Users\radek> npm install mongoose
```

Database Access

Database Users Custom Roles

+ ADD NEW DATABASE USER

User Name	Authentication Method	MongoDB Roles	Resources	Actions
radoslaw	SCRAM	readWriteAnyDatabase@admin	All Resources	<button>EDIT</button> <button>DELETE</button>

Add entries to your IP Access List

Only an IP address you add to your Access List will be able to connect to your project's clusters. You can manage existing IP entries via the [Network Access Page](#).

IP Address	Description
Enter IP Address	Enter description
Add My Current IP Address	
Add Entry	

This IP address has already been added.

IP Access List	Description
77.255.107.45/32	My IP Address

[!\[\]\(ab20479487461795fab4bd828006ec9a_img.jpg\) EDIT](#) [!\[\]\(6cd7fe98f921a38e52df30f8d3407774_img.jpg\) REMOVE](#)

[Finish and Close](#)

Connect to IIMap

Set up connection security Choose a connection method Connect

Connecting with MongoDB Driver

1. Select your driver and version

We recommend installing and using the latest driver version.

Driver	Version
Node.js	5.5 or later

2. Install your driver

Run the following on the command line

```
npm install mongodb
```

[View MongoDB Node.js Driver installation instructions.](#)

3. Add your connection string into your application code

View full code sample

```
mongodb+srv://radoslaw:<password>@iimap.pvihhyy.mongodb.net/?retryWrites=true&w=majority
```

```
const express = require('express');
const { MongoClient, ServerApiVersion } = require('mongodb');
const bodyParser = require('body-parser');
```

```
const uri = "mongodb+srv://radoslaw:<pass***word>iimap.
```

```
// Stworzenie klienta
const client = new MongoClient(uri, {
  serverApi: {
    version: ServerApiVersion.v1,
    strict: true,
    deprecationErrors: true,
  }
});
```

```
async function run() {
    try {
        // Połączenie klienta z serwerem
        await client.connect();

        // Ping połączenia z serwerem
        await client.db("IIImap").command({ ping: 1 });
        console.log("Pinged your deployment. You successfully connected to MongoDB!");

    } finally {
        // Zamknięcie klienta
        await client.close();
    }
} // wyświetlanie błędów
run().catch(console.dir);
```

```
async function getData() {  
  let result = await client.connect();  
  db = result.db('IImap');  
  return db.collection('products');  
}
```

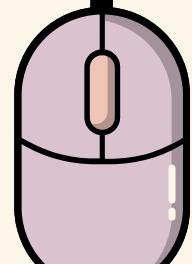
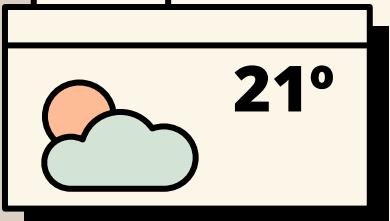
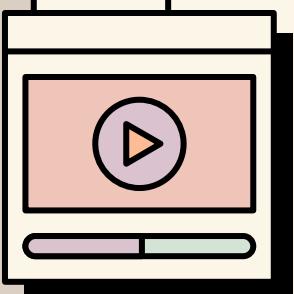
```
dbConnect().then((resp)=>{
  resp.find({name: 'nord'}).toArray().then((data)=>{
    console.log(data);
  });
})
```

Koniec

Połączenie aplikacji z bazą danych

Relacyjna baza danych

>>>



Użyte technologie



PHP	Do połączenia aplikacji z bazą danych
MySql	Baza danych
HTML, CSS	Struktura strony

PHP

PHP jest skryptowym językiem programowania, wykorzystywanym najczęściej do tworzenia stron internetowych.

Wykonywany jest po stronie serwera, co oznacza, że jest niewidoczny dla użytkowników. Pozwala na m.in. dynamiczne generowanie strony, tworzenie i modyfikowanie plików na serwerze, ograniczanie dostępu do danych podstron na twojej stronie, szyfrowanie danych czy stosowanie ciasteczek

Relacyjna baza danych

Relacyjna baza danych to rodzaj bazy danych, który pozwala przechowywać powiązane ze sobą elementy danych i zapewnia do nich dostęp. Relacyjne bazy danych są oparte na modelu relacyjnym — jest to prosty i intuicyjny sposób przedstawiania danych w tabelach. W relacyjnej bazie danych każdy wiersz tabeli jest rekordem z unikatowym identyfikatorem nazywanym kluczem.

Obsługa bazy

Aby administrować bazą danych, użytkownicy komunikują się z interfejsem zarządzania za pomocą języka SQL, który jest oparty na algebrze relacyjnej. Wszystkie polecenia są wyrażone w języku SQL i szczegółowo opisują każdy etap tworzenia bazy danych. Język ten umożliwia wybór pożądanych informacji, wskazywanie ich lokalizacji w bazie danych, interakcje między nimi, etc.



Zalety oraz wady

Zalety	Wady
Model tabelaryczny	Nie wszystko da się zamieścić w takich bazach np.: dane abstrakcyjne
Spójność danych	Dane nie mogą być hierarchizowane
Znormalizowany język zapytań	Segmentacja danych

Połączenie z bazą za pomocą PHP

Do uruchomienia należy pobrać program XAMPP.
Oraz uruchomić usługi Apache, MySQL.



Baza do łączenia

Będę łączyć się z bazą danych która wygląda tak.

Prosta baza danych z 3 kolumnami. ID, Nazwa Miasta oraz Ludność

	← T →		id	nazwa	ludnosc			
<input type="checkbox"/>		Edytuj		Kopiuj		Usuń	1 Kraków	800000
<input type="checkbox"/>		Edytuj		Kopiuj		Usuń	2 Warszawa	1000000
<input type="checkbox"/>		Edytuj		Kopiuj		Usuń	3 Poznań	600000

Skrypt

Tworzymy plik index.PHP
pozwalający nam na
zamieszczenie kodu PHP w
strukturze HTML.



Skrypt obok łączy się z naszą bazą danych która nazywa się „`baza_danych`”. Pobiera informacje o miastach i wypisuje o nich informacje.
Posiada obsługę połączenia.

```
<?php

// Ustawienia do połączenia z bazą danych
$host = "localhost";
$user = "root";

$database = "baza_danych";

// Nawiązanie połączenia
$connection = new mysqli($host, $user, '', $database);

// Sprawdzenie połączenia
if ($connection->connect_error) {
    die("Connection failed: " . $connection->connect_error);
}

// Zapytanie SQL
$query = "SELECT * FROM miasta";

// Wykonanie zapytania
$result = $connection->query($query);

// Obsługa wyników
if ($result->num_rows > 0) {
    // Wyświetlenie wyników
    while ($row = $result->fetch_assoc()) {
        echo "ID: " . $row["id"] . ", Nazwa: " . $row["nazwa"] . ", Ludność: " . $row["ludosc"] . "<br>";
    }
} else {
    echo "Brak wyników.";
}
echo "Zostało nawiązane połączenie!";

// Zamknięcie połączenia
$connection->close();
?>
```

Dziękuje za uwagę