



# Database

# Database

DATABASE(baza danych) – uporządkowany zbiór danych zapisany zgodnie z określonymi regułami.

Definicja obejmuje dane cyfrowe gromadzone zgodnie z zasadami przyjętymi dla danego programu komputerowego specjalizowanego do gromadzenia i przetwarzania danych.

# Database

Program taki nazywany jest systemem zarządzania bazami danych(DBMS, database management system), składający się ze zintegrowanego zestawu oprogramowania komputerowego, które pozwala użytkownikom na interakcję z jedną lub większą liczbą baz danych i zapewnia dostęp do wszystkich danych zawartych w bazie danych

# **DZIAŁANIE BAZ DANYCH**

Sposób organizacji danych w bazach danych jest różny.

Sposób organizacji danych w bazach danych jest różny.

Wiele typów baz danych organizuje dane w formie tabel zawierających rekordy podzielone na pola, w których są przechowywane informacje poszczególnych kategorii.

# RODZAJE BAZ DANYCH

Bazy danych można podzielić według struktur organizacji danych, których używają:

- Bazy proste:
  - kartotekowe
  - hierarchiczne
- Bazy złożone:
  - relacyjne
  - obiektowe
  - relacyjno - obiektowe
  - strumieniowe
  - nierelacyjne (NoSQL)

# **BAZY KARTOTEKOWE**

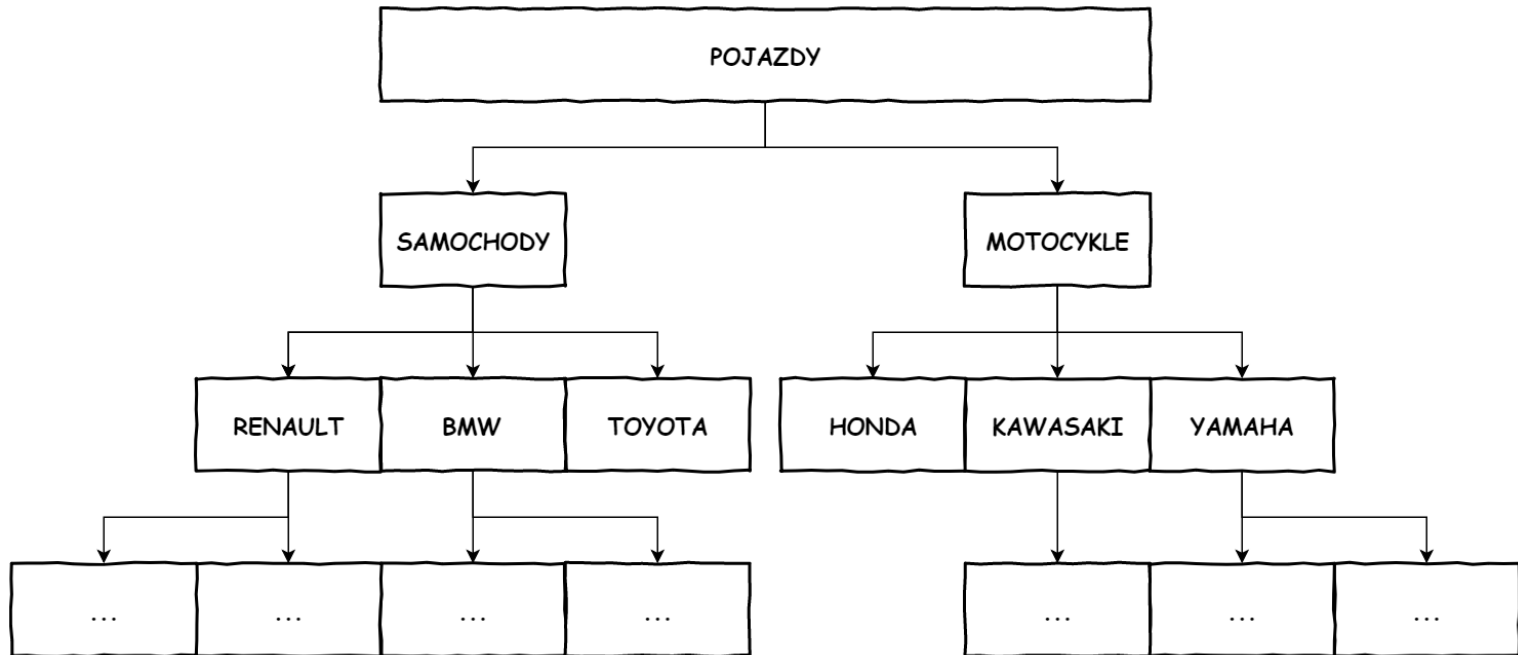
W bazach kartotekowych każda tablica danych jest samodzielnym dokumentem. Jedna tablica nie może współpracować z innymi. Przykładami tego typu bazy danych mogą być: książki telefoniczne, kucharskie. Są one stosowane w jednym, wybranym wcześniej celu.

| Id | FirstName | LastName    | Email                                      | Phone | Street       | City      | ZipCode |  |
|----|-----------|-------------|--|-------|--------------|-----------|---------|--|
| 1  | Adam      | Mickiewicz  | <a href="mailto:adam@m.com">adam@m.com</a> | 1234  | Lipowa       | Białystok | 12-345  |  |
| 2  | Jan       | Nowak       | <a href="mailto:jan@n.com">jan@n.com</a>   |       | Krasnoluda   | Zęwągi    | 23-456  |  |
| 3  | Anna      | Adamek      | <a href="mailto:a@a.com">a@a.com</a>       | 8764  | Sienkiewicza | Białystok | 45-123  |  |
| 4  | Jan       | Kowalski    | <a href="mailto:j@k.com">j@k.com</a>       |       |              |           |         |  |
| 5  | Zenon     | Burczymucha | <a href="mailto:z@b.com">z@b.com</a>       |       | Smerfa       | Lublin    | 88-213  |  |
| 6  | ...       | ...         | ...  | ...   | ...          | ...       | ...     |  |
|    |           |             |  |       |              |           |         |  |
|    |           |             |  |       |              |           |         |  |



# **BAZY HIERARCHICZNE**

Są to bazy danych zawierające powiązane ze sobą dane, ułożone w strukturę o kształcie drzewa z jednym punktem początkowym i wieloma rozgałęzieniami. Hierarchiczny model bazy danych cechuje skomplikowana struktura i trudności z tworzeniem określonego typu powiązań pomiędzy danymi.



# BAZY RELACYJNE

W przeciwieństwie do baz kartotekowych, w bazach relacyjnych wiele tablic danych może ze sobą współpracować.

Bazy te posiadają wewnętrzne języki programowania, wykorzystujące zwykłe SQL do operowania na danych, dzięki czemu tworzone są zaawansowane funkcje obsługi danych.

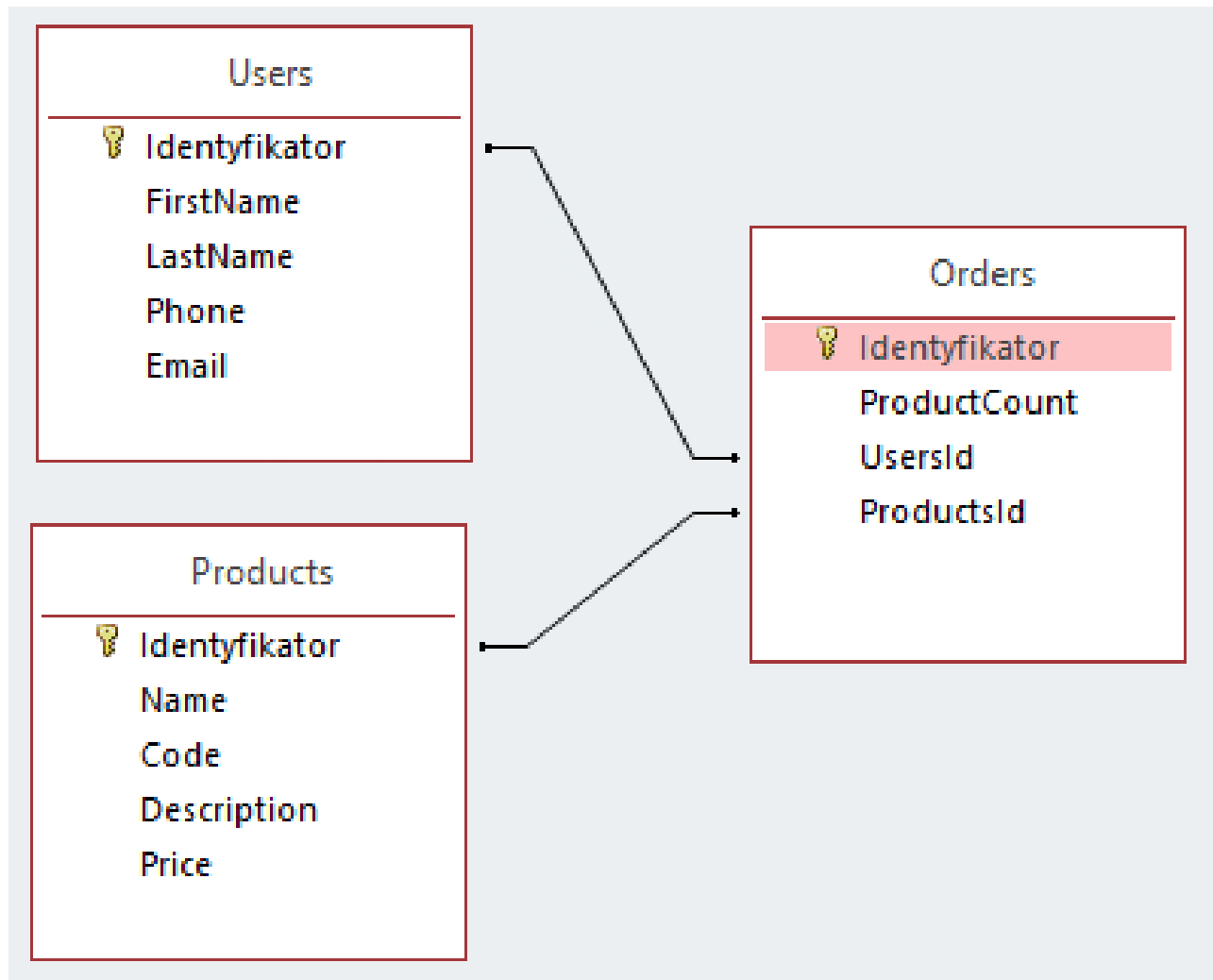
# BAZY RELACYJNE

Relacyjne bazy danych oparte są na kilku podstawowych zasadach:

- Wszystkie wartości są oparte na prostych typach danych
- Wszystkie dane w bazie relacyjnej przedstawiane są w formie tabel. Tabela składa się z wierszy zwanych rekordami i kolumn zwanych polami. Każda kolumna posiada unikalną w obrębie tabeli nazwę

# BAZY RELACYJNE

- Po wprowadzeniu danych do bazy, możliwe jest porównywanie wartości z różnych kolumn, również z różnych tabel, scalanie wierszy, gdy pochodzące z nich wartości są tego samego typu.
- Każda tabela musi posiadać jedną kolumnę pozwalającą jednoznacznie zidentyfikować i odnaleźć konkretny wiersz. Kolumnę tę określa się mianem 'klucza podstawowego tabeli'



# **BAZY OBIEKTOWE**

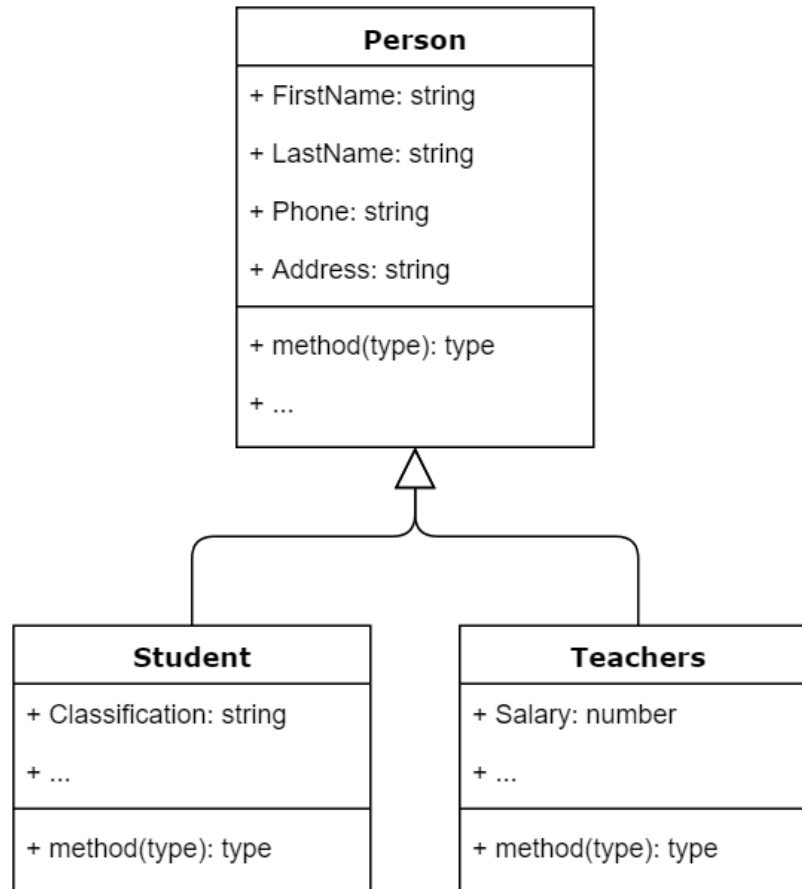
W bazach obiektowych dane są przechowywane za pomocą struktur obiektowych, będących zdefiniowanych jako klasy. Nie są one zdefiniowane żadnym oficjalnym standardem.

# **BAZY OBIEKTOWE**

Cechą charakterystyczną obiektowych baz danych jest to, że przechowują obiekty o dowolnych strukturach wraz z przywiązanymi do nich metodami (procedurami).

Dzięki takiemu rozwiązaniu bazy takie mają znaczną przewagę nad innymi rodzajami baz kiedy zachodzi konieczność przechowywania bardzo złożonych struktur.





# **BAZY RELACYJNO- OBIEKTOWE**

Bazy relacyjno-obiektowe pozwalają na manipulowanie danymi jako zestawem obiektów, posiadają jednak bazę relacyjną jako wewnętrzny mechanizm przechowywania danych.

# **BAZY STRUMIENIOWE**

Strumieniowe bazy danych są to bazy, w których dane są przedstawione w postaci strumieni danych. System zarządzania taką bazą nazywany jest DSMS (ang. Data Stream Managament System) czyli strumieniowym systemem zarządzania danymi.

# **BAZY STRUMIENIOWE**

W modelu strumieni danych zakłada się, że niektóre lub wszystkie napływające do systemu dane nie są dostępne w dowolnej chwili. Możliwy czas, w jakim można je zarejestrować jest skończony. Dane te pojawiają się w źródle danych i przyjmują postać strumienia danych.

# BAZY NIERELACYJNE

Pod pojęciem bazy nierelacyjnej (NoSQL database) najczęściej rozumie się przechowywanie danych w formie listy par obiektów klucz-wartość, w których nie występują powiązania relacyjne między przechowywanymi obiektami.

W bazie NoSQL najczęściej nie ma wymagania aby obiekty były jednorodne pod względem struktury.

## Users

```
{  
  _id: "12345...",  
  firstName: "Jan",  
  lastName: "Nowak",  
  coordinates: {  
    latitude: 12.34,  
    longitude: 141.21  
  }  
}
```

```
{  
  _id: "12345...",  
  firstName: "Adam",  
  lastName: "Mickiewicz",  
  coordinates: {  
    latitude: 52.34,  
    longitude: 41.21  
  }  
}
```

# SQL VS NOSQL

# SQL

SQL (Structured Query Language) – strukturalny język zapytań używany do tworzenia, modyfikowania baz danych oraz do umieszczania i pobierania danych z baz danych.

Język SQL jest językiem deklaratywnym. Decyzję o sposobie przechowywania i pobrania danych pozostawia się systemowi zarządzania bazą danych (DBMS).





```
1 SELECT * FROM EMPLOYEES
2     WHERE SALARY > 2000
3     ORDER BY SALARY DESC
```

# NoSQL

NoSQL (nierelacyjna baza danych SQL) – baza danych zapewniająca mechanizm do przechowywania i wyszukiwania danych modelowanych w inny sposób niż relacje tabelaryczne używane w relacjach baz danych SQL.

NoSQL stworzony został z potrzeby obsługiwanie większych wolumenów danych, która wymusiła przejście na model budowania platform na klastrach mniej wydajnych serwerów.

# NoSQL

Struktury danych używane przez NoSQL (np. klucz-wartość, graf, dokument, szerokokolumnowe) różnią się od tych używanych domyślnie w relacyjnych bazach danych, dzięki czemu niektóre operacje NoSQL są szybsze.

**DLACZEGO NoSQL?**



[illegible]

[illegible]

⏪
⏩
a1
a2
**a3**
b\_users
b\_phones
b\_address
⊕
⋮
⏪
⏩





Autozapis

db.ods - Zapisano

Piotr Tarasiuk

Plik

Narzędzia główne

Wstawianie

Rysowanie

Układ strony

Formuły

Dane

Recenzja

Widok

Pomoc

Zespół

Powiedz mi, co chcesz zrobić

Udostępnij

A1

X

✓

f<sub>x</sub>

UserId

|    | A      | B     | C       | D | E | F | G | H | I | J | K | L | M | N | O |
|----|--------|-------|---------|---|---|---|---|---|---|---|---|---|---|---|---|
| 1  | UserId | Phone | Label   |   |   |   |   |   |   |   |   |   |   |   |   |
| 2  | 1      | 1234  | private |   |   |   |   |   |   |   |   |   |   |   |   |
| 3  | 3      | 8764  | private |   |   |   |   |   |   |   |   |   |   |   |   |
| 4  | 1      | 7473  | work    |   |   |   |   |   |   |   |   |   |   |   |   |
| 5  | 2      | 2341  | work    |   |   |   |   |   |   |   |   |   |   |   |   |
| 6  | 3      | 6423  | work    |   |   |   |   |   |   |   |   |   |   |   |   |
| 7  | 4      | 3352  | work    |   |   |   |   |   |   |   |   |   |   |   |   |
| 8  | 5      | 1247  | work    |   |   |   |   |   |   |   |   |   |   |   |   |
| 9  |        |       |         |   |   |   |   |   |   |   |   |   |   |   |   |
| 10 |        |       |         |   |   |   |   |   |   |   |   |   |   |   |   |
| 11 |        |       |         |   |   |   |   |   |   |   |   |   |   |   |   |
| 12 |        |       |         |   |   |   |   |   |   |   |   |   |   |   |   |
| 13 |        |       |         |   |   |   |   |   |   |   |   |   |   |   |   |
| 14 |        |       |         |   |   |   |   |   |   |   |   |   |   |   |   |
| 15 |        |       |         |   |   |   |   |   |   |   |   |   |   |   |   |
| 16 |        |       |         |   |   |   |   |   |   |   |   |   |   |   |   |
| 17 |        |       |         |   |   |   |   |   |   |   |   |   |   |   |   |
| 18 |        |       |         |   |   |   |   |   |   |   |   |   |   |   |   |
| 19 |        |       |         |   |   |   |   |   |   |   |   |   |   |   |   |

<

>

a1

a2

a3

b\_users

**b\_phones**

b\_address

+

Gotowy

-

+

150%

[illegible]

db.ods - Zapisano

Piotr Tarasiuk

Plik Narzędzia główne Wstawianie Rysowanie Układ strony Formuły Dane Recenzja Widok Pomoc Zespół Powiedz mi, co chcesz zrobić Udostępnij

A1 X ✓ fx UserId

|    | A             | B                | C               | D | E | F | G | H | I | J | K | L | M | N |
|----|---------------|------------------|-----------------|---|---|---|---|---|---|---|---|---|---|---|
| 1  | <b>UserId</b> | <b>FirstName</b> | <b>LastName</b> |   |   |   |   |   |   |   |   |   |   |   |
| 2  | 1             | Adam             | Mickiewicz      |   |   |   |   |   |   |   |   |   |   |   |
| 3  | 2             | Jan              | Nowak           |   |   |   |   |   |   |   |   |   |   |   |
| 4  | 3             | Anna             | Adamek          |   |   |   |   |   |   |   |   |   |   |   |
| 5  | 4             | Jan              | Kowalski        |   |   |   |   |   |   |   |   |   |   |   |
| 6  | 5             | Zenon            | Burczymucha     |   |   |   |   |   |   |   |   |   |   |   |
| 7  | 6             | ...              | ...             |   |   |   |   |   |   |   |   |   |   |   |
| 8  | Users         |                  |                 |   |   |   |   |   |   |   |   |   |   |   |
| 9  |               |                  |                 |   |   |   |   |   |   |   |   |   |   |   |
| 10 |               |                  |                 |   |   |   |   |   |   |   |   |   |   |   |
| 11 |               |                  |                 |   |   |   |   |   |   |   |   |   |   |   |
| 12 |               |                  |                 |   |   |   |   |   |   |   |   |   |   |   |
| 13 |               |                  |                 |   |   |   |   |   |   |   |   |   |   |   |
| 14 |               |                  |                 |   |   |   |   |   |   |   |   |   |   |   |
| 15 |               |                  |                 |   |   |   |   |   |   |   |   |   |   |   |
| 16 |               |                  |                 |   |   |   |   |   |   |   |   |   |   |   |
| 17 |               |                  |                 |   |   |   |   |   |   |   |   |   |   |   |
| 18 |               |                  |                 |   |   |   |   |   |   |   |   |   |   |   |
| 19 |               |                  |                 |   |   |   |   |   |   |   |   |   |   |   |

Addresses

| UserId | Street       | City      | ZipCode | Label |
|--------|--------------|-----------|---------|-------|
| 1      | Lipowa       | Białystok | 12-345  | home  |
| 2      | Krasnoluda   | Zęwągi    | 23-456  | home  |
| 3      | Sienkiewicza | Białystok | 45-123  | home  |
| 5      | Smerfa       | Lublin    | 88-213  | home  |
| 2      | Reduta       | Warszawa  | 12-555  | mail  |
| 4      | Mars         | Gdańsk    | 77-222  | mail  |

Phones

| UserId | Phone | Label   |
|--------|-------|---------|
| 1      | 1234  | private |
| 3      | 8764  | private |
| 1      | 7473  | work    |
| 2      | 2341  | work    |
| 3      | 6423  | work    |
| 4      | 3352  | work    |
| 5      | 1247  | work    |

a1 a2 a3 b\_users b\_phones b\_address

Gotowy

150%

# MONGODB

MongoDB – otwarty, nierelacyjny system zarządzania bazą danych. Charakteryzuje się dużą skalowalnością, wydajnością oraz brakiem ściśle zdefiniowanej struktury obsługiwanych baz danych.

Zamiast tego dane składowane są jako dokumenty w stylu JSON, co umożliwia aplikacjom bardziej naturalne ich przetwarzanie, przy zachowaniu możliwości tworzenia hierarchii oraz indeksowania.

**SQL**

**NoSQL**

**SQL**

**DATABASE**



**NoSQL**

**DATABASE**



**SQL**

**DATABASE**



**NoSQL**

**DATABASE**



| id  | first name | last name |
|-----|------------|-----------|
| 1   | Jan        | Nowak     |
| 2   | ...        | ...       |
| 5   | ...        | ...       |
| ... | ...        | ...       |



## SQL

DATABASE



| id  | first name | last name |
|-----|------------|-----------|
| 1   | Jan        | Nowak     |
| 2   | ...        | ...       |
| 5   | ...        | ...       |
| ... | ...        | ...       |

## NoSQL

DATABASE



```
[
  {
    "id": "abcs23...",
    "firstName": "Jan",
    "lastName": "Nowak",
  },
  {
    ...
  }
]
```

## SQL

DATABASE



TABLE



| id  | first name | last name |
|-----|------------|-----------|
| 1   | Jan        | Nowak     |
| 2   | ...        | ...       |
| 5   | ...        | ...       |
| ... | ...        | ...       |

## NoSQL

DATABASE



```
[  
  {  
    "id": "abcs23...",  
    "firstName": "Jan",  
    "lastName": "Nowak",  
  },  
  {  
    ...  
  }  
]
```

## SQL

DATABASE



TABLE



| id  | first name | last name |
|-----|------------|-----------|
| 1   | Jan        | Nowak     |
| 2   | ...        | ...       |
| 5   | ...        | ...       |
| ... | ...        | ...       |

## NoSQL

DATABASE



COLLECTION



```
[  
  {  
    "id": "abcs23...",  
    "firstName": "Jan",  
    "lastName": "Nowak",  
  },  
  {  
    ...  
  }  
]
```

## SQL

DATABASE



TABLE

ROW

| id  | first name | last name |
|-----|------------|-----------|
| 1   | Jan        | Nowak     |
| 2   | ...        | ...       |
| 5   | ...        | ...       |
| ... | ...        | ...       |

## NoSQL

DATABASE



COLLECTION

```
[
  {
    "id": "abcs23...",
    "firstName": "Jan",
    "lastName": "Nowak",
  },
  {
    ...
  }
]
```

## SQL

DATABASE



TABLE

ROW

| id  | first name | last name |
|-----|------------|-----------|
| 1   | Jan        | Nowak     |
| 2   | ...        | ...       |
| 5   | ...        | ...       |
| ... | ...        | ...       |

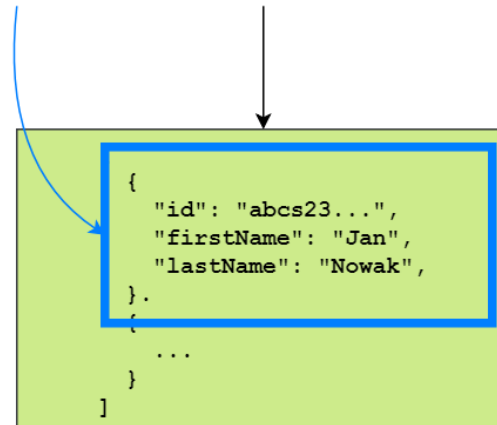
## NoSQL

DATABASE



COLLECTION

DOCUMENT



## SQL

DATABASE



ROW

TABLE

COLUMN

| id  | first name | last name |
|-----|------------|-----------|
| 1   | Jan        | Nowak     |
| 2   | ...        | ...       |
| 5   | ...        | ...       |
| ... | ...        | ...       |

## NoSQL

DATABASE



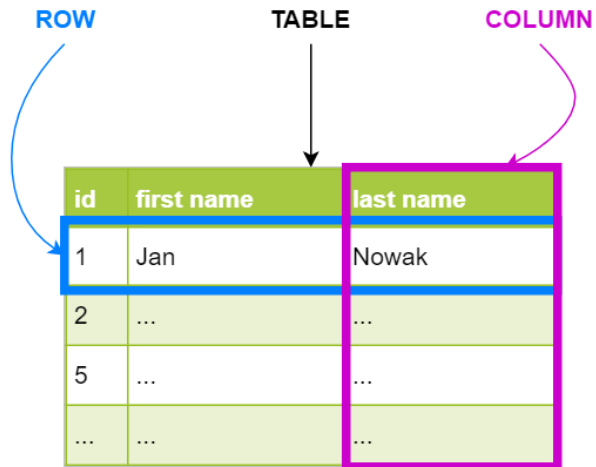
DOCUMENT

COLLECTION

|   |
|---|
| <pre>{   "id": "abcs23...",   "firstName": "Jan",   "lastName": "Nowak", }. {   ... } ]</pre> |
|---|

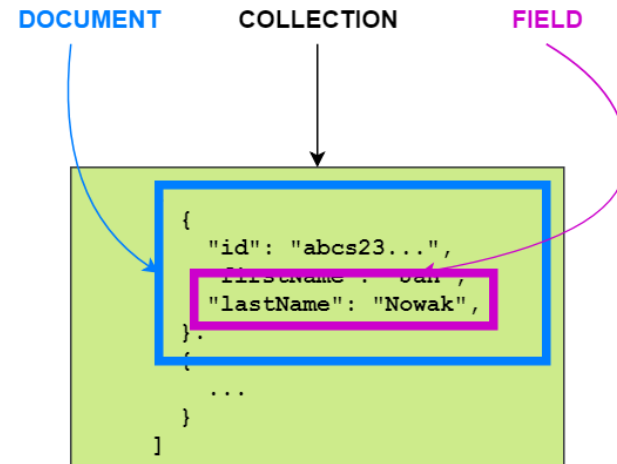
## SQL

DATABASE



## NoSQL

DATABASE



# **MongoDB console**



```
C:\Windows\System32\cmd.exe - mongo
D:\mongodb\bin>mongo
MongoDB shell version v4.0.8
connecting to: mongodb://127.0.0.1:27017/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("6eaecc5c-ac26-49b4-ae33-e12177a20919") }
MongoDB server version: 4.0.8
Server has startup warnings:
2019-04-04T06:12:41.333-0700 I CONTROL [initandlisten]
2019-04-04T06:12:41.333-0700 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2019-04-04T06:12:41.333-0700 I CONTROL [initandlisten] **           Read and write access to data and configuration is unrestricted.
2019-04-04T06:12:41.333-0700 I CONTROL [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you and anyone you share the URL with. MongoDB may use this information to make product improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---

> show dbs
abc      0.000GB
admin    0.000GB
config   0.000GB
local    0.000GB
>
```

# **MongoDB Compass**

MongoDB Compass Community - localhost:27017

ConnectViewHelp

My Cluster

localhost:27017STANDALONE

MongoDB 4.0.8 Community

4 DBS3 COLLECTIONS

filter

abcadminconfiglocal

+

Databases

CREATE DATABASE

| Database Name | Storage Size | Collections | Indexes |  |
|---------------|--------------|-------------|---------|--|
| abc           | 40.0KB       | 2           | 2       |  |
| admin         | 16.0KB       | 0           | 1       |  |
| config        | 24.0KB       | 0           | 2       |  |
| local         | 16.0KB       | 1           | 1       |  |

# **MongoDB in NodeJS**



```
1  const { MongoClient } = require('mongodb');
2
3  (async () => {
4      const dbAddress = 'mongodb://127.0.0.1:27017/';
5      const dbName = 'users';
6      const mongoOptions = { useNewUrlParser : true };
7      const client = await MongoClient.connect(dbAddress, mongoOptions);
8
9      const db = client.db(dbName);
10     // ...
11 })();
```

**<https://docs.mongodb.com/drivers/node/>**