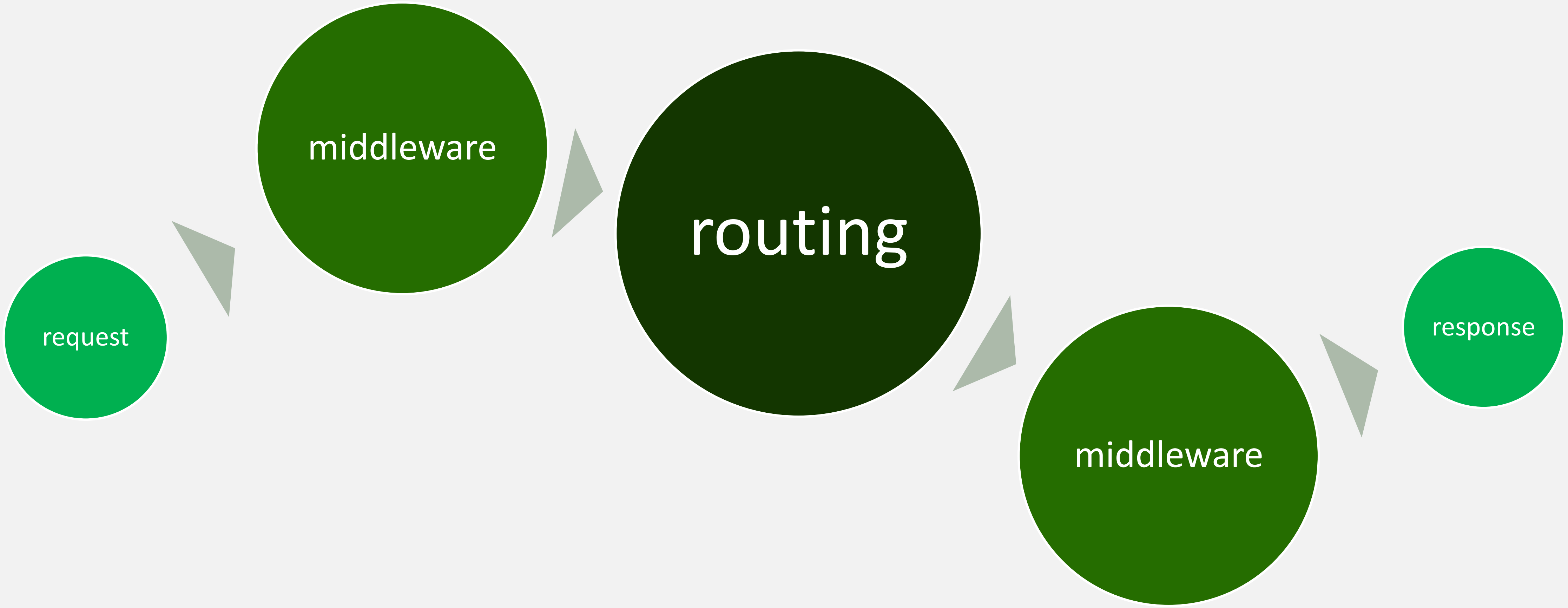
NodeJS
środowisko i technologia ServerSide

PAWEŁ ŁUKASZUK


request
middleware
routing
middleware
response

71
HEAT SET TO
63

Environment variables

Typically, our applications require many variables to be set in order for them to work. By relying on external configurations, your app can easily be deployed on different environments.

These changes are independent of code changes, so they do not require your application to be rebuilt to change.

Data which changes depending on the environment your app is running on should be set as environment variables.

Environment variables should be stored outside of code ouf our app!

Examples

Some common examples:

- HTTP Port and Address
- Database, cache, and other storage connection information
- Location of static files/folders
- Endpoints of external services
- Sensitive data like API keys should not be in the source code, or known to persons who do not need access to those external services.

dotenv library

This library does one simple task: loads variables from a .env file into the process.env object

https://www.npmjs.com/package/dotenv

https://github.com/motdotla/dotenv#readme

```
# .env file
myvar1=test
```

```
// app.js file
require('dotenv').config();
console.log(process.env.myvar1);
// test
```

Rules of environment variables

never commit variables to the source code repository

- add .env file to .gitginore
- if you need to share information about variables you can:
  - create file .sample-env with variables’ names only
  - describe necessary variables in readme file or other documentation
- in application code don’t modify variables that was already set

Kolonie
Komenský, J. A. pedagogik
Komsomol
Kopaná ČSSR
Korporace

Database

Database - an organized collection of data stored according to specific rules.

The definition includes digital data collected according to the rules adopted for a particular computer program specialized for data collection and processing.

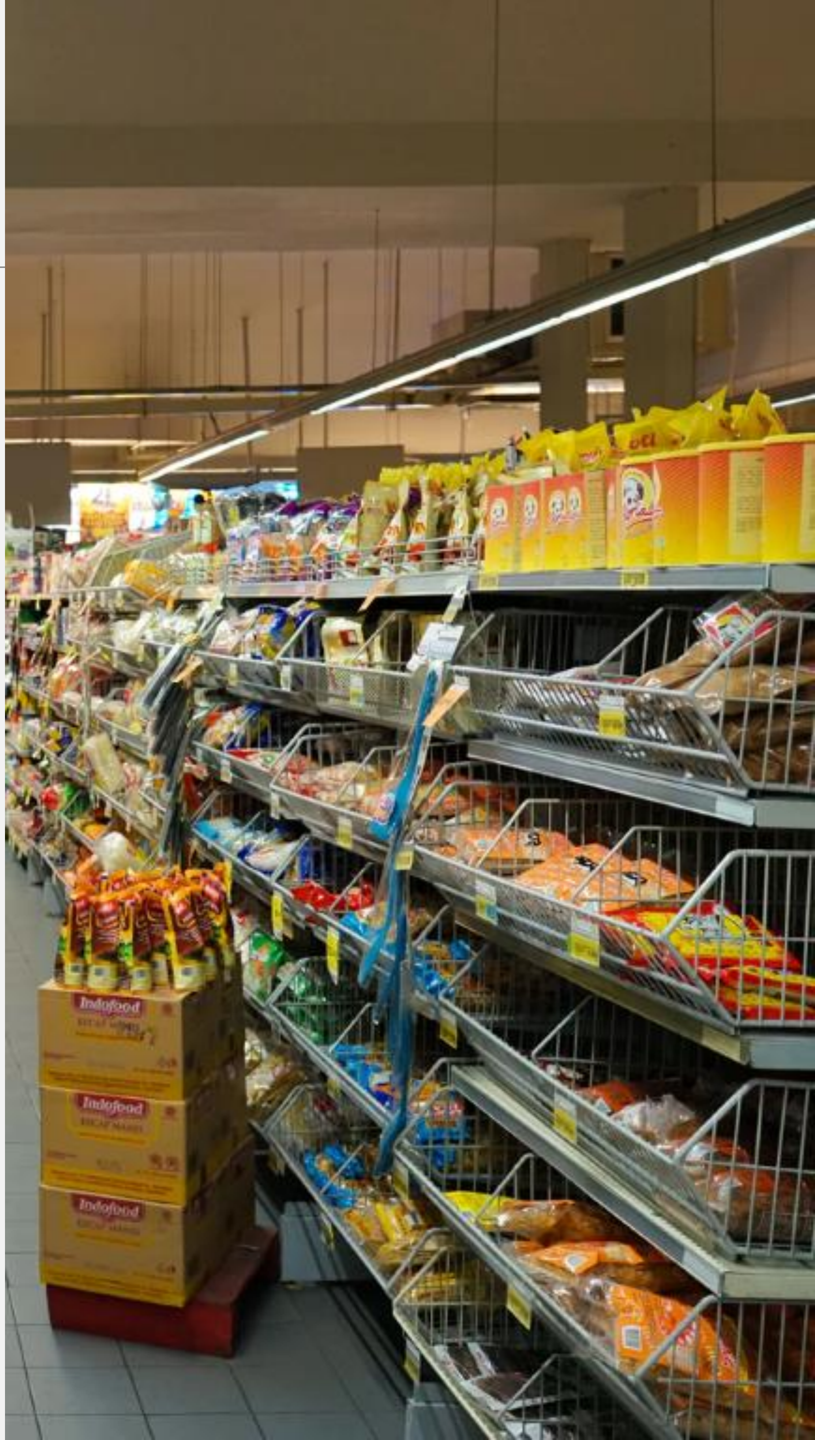
Database Management System

Database management system (DBMS) - consists of an integrated set of computer software that allows users to interact with one or more databases and provides access to all the data contained in the database

Database organization

Data can be organizaed in database in many different ways.

Many types of databases organize data in the form of tables containing records divided into fields that store information of particular categories.



Types of databases

Databases can be divided according to the data organization structures they use:

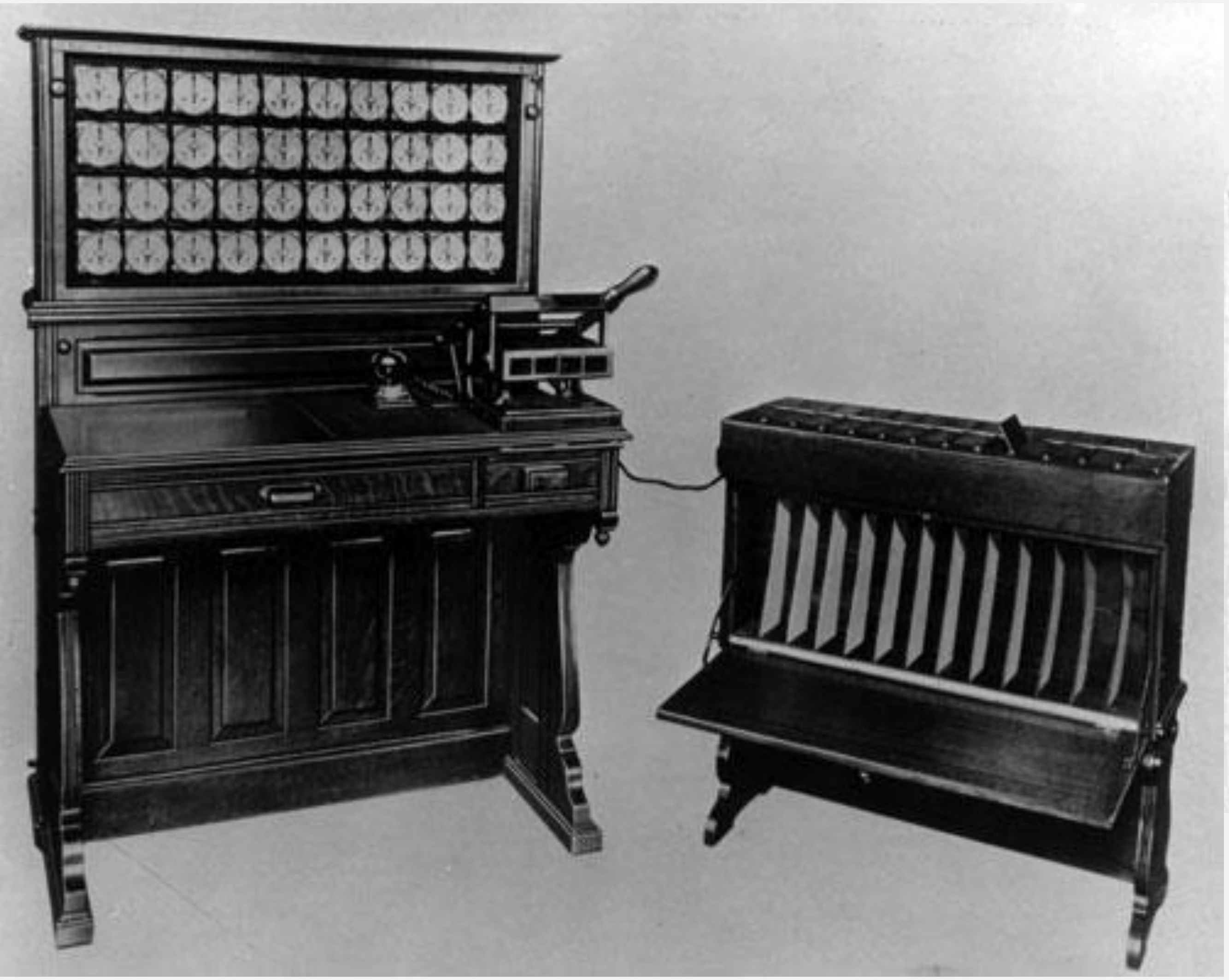
- simple databases:
  - card-based
  - hierarchical
- complex databases:
  - relational (SQL)
  - object-oriented
  - relational-object-oriented
  - streaming
  - non-relational (NoSQL)

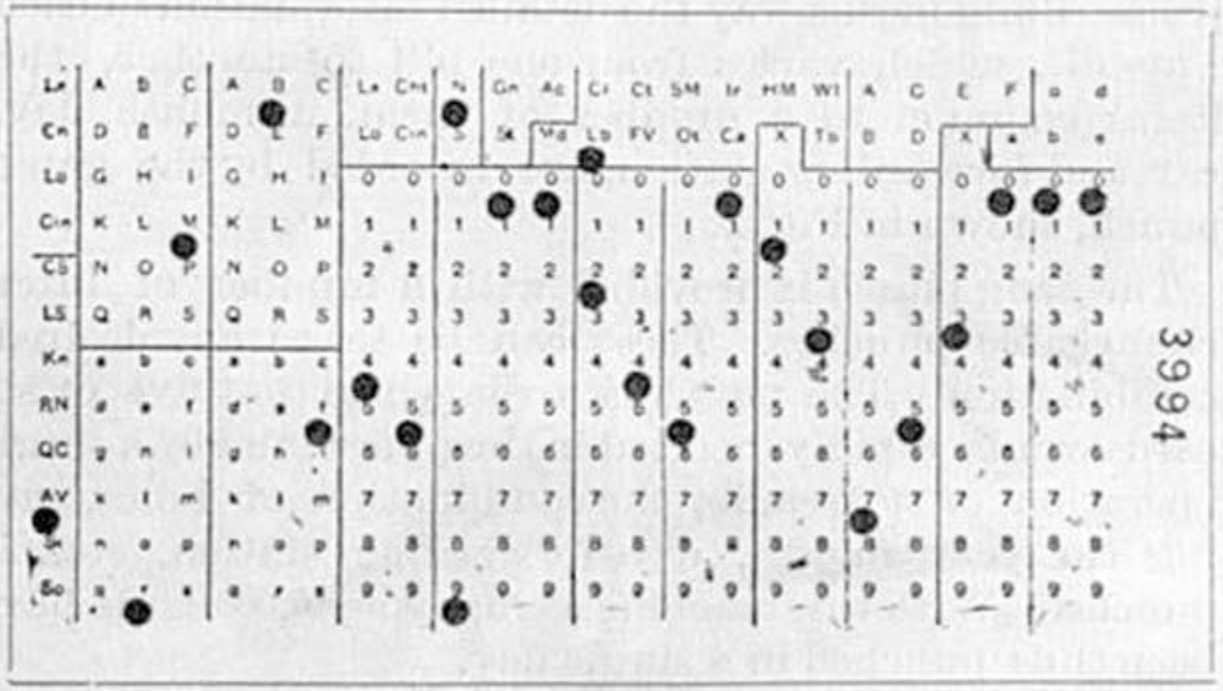
Card-based databases

In card-base databases, each data card is a stand-alone document.

One document cannot cooperate with others.

They are used for a single, pre-selected purpose.

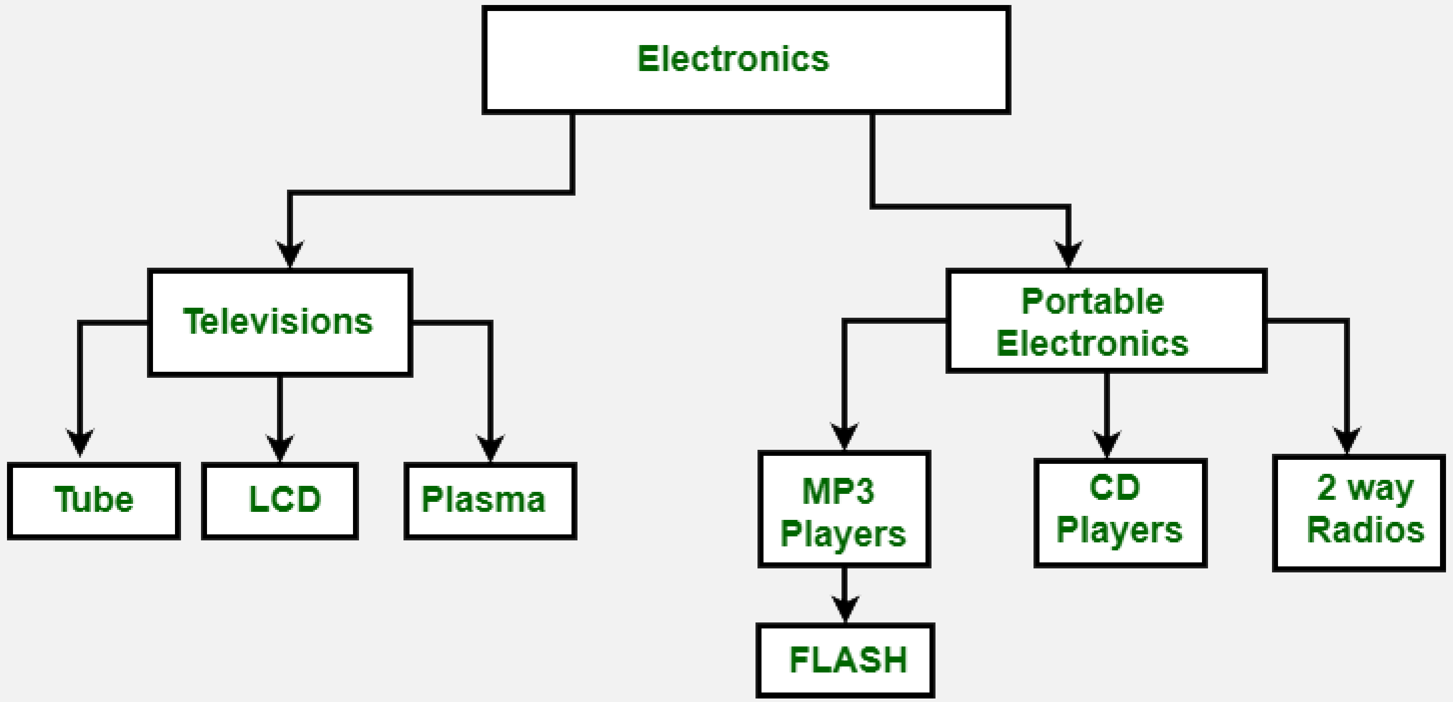



3994

Hierarchical databases

These databases are containing related data, arranged in a tree-like structure with one starting point and many branches.

Hierarchical database model is characterized by complex structure and difficulty in creating relationships between data.



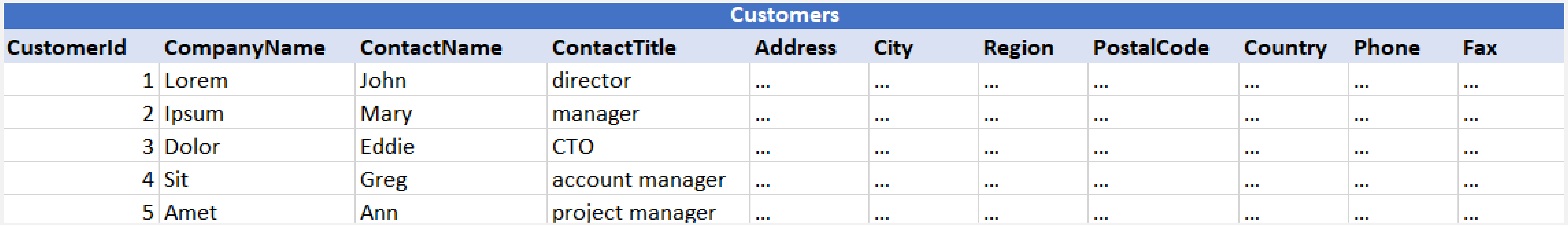
Relational databases

Relational databases are based on several basic principles:

- values are based on simple data types
- data in a relational database are represented as tables.

Table consists of rows called records and columns called fields.
Each column has a name unique within the table.

Each table must have one column to uniquely identify and find a particular row.
This column is referred to as the „primary key of the table”.



Customers

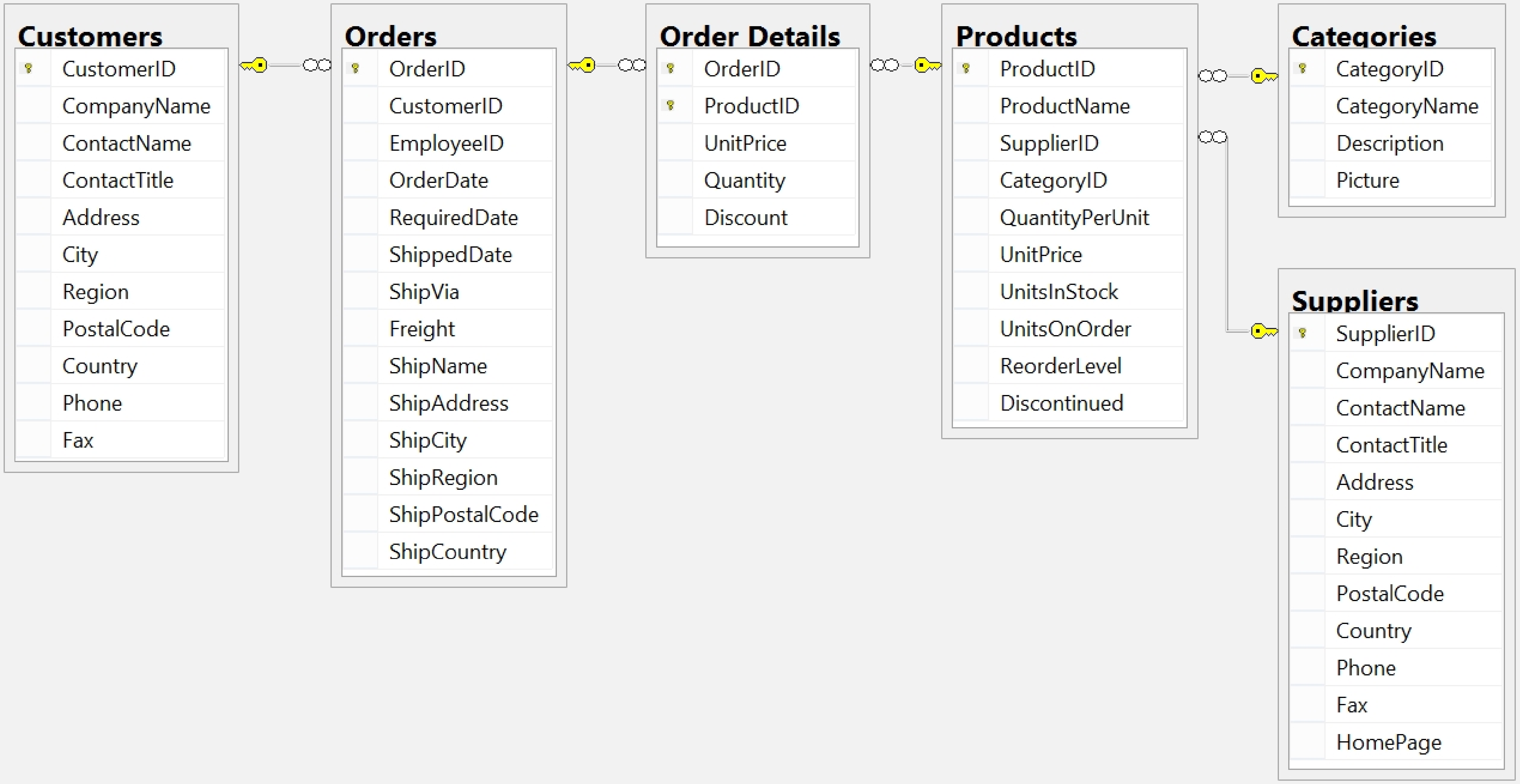
CustomerId	CompanyName	ContactName	ContactTitle	Address	City	Region	PostalCode	Country	Phone	Fax
1	Lorem	John	director	...	...	...	...	...	...	...
2	Ipsum	Mary	manager	...	...	...	...	...	...	...
3	Dolor	Eddie	CTO	...	...	...	...	...	...	...
4	Sit	Greg	account manager	...	...	...	...	...	...	...
5	Amet	Ann	project manager	...	...	...	...	...	...	...

Relational databases #2

Unlike card databases, in relational databases multiple data tables can work together.

Once the data is entered into the database, it is possible to compare values from different columns, even from different tables, merging rows when their values are of the same type.

These databases have internal programming languages to operate on the data (usually based on SQL language).



Customers
CustomerID
CompanyName
ContactName
ContactTitle
Address
City
Region
PostalCode
Country
Phone
Fax
Orders
OrderID
CustomerID
EmployeeID
OrderDate
RequiredDate
ShippedDate
ShipVia
Freight
ShipName
ShipAddress
ShipCity
ShipRegion
ShipPostalCode
ShipCountry
Order Details
OrderID
ProductID
UnitPrice
Quantity
Discount
Products
ProductID
ProductName
SupplierID
CategoryID
QuantityPerUnit
UnitPrice
UnitsInStock
UnitsOnOrder
ReorderLevel
Discontinued
Categories
CategoryID
CategoryName
Description
Picture
Suppliers
SupplierID
CompanyName
ContactName
ContactTitle
Address
City
Region
PostalCode
Country
Phone
Fax
HomePage

Structured Query Language

SQL (Structured Query Language) - a structured query language used to create, modify databases and to place and retrieve data from databases.

SQL is a declarative language - decision on how to store and retrieve data is left to the database management system (DBMS).

```
SELECT PRODUCTNAME, UNITPRICE
     FROM PRODUCTS
WHERE PRICE > 2000
ORDER BY PRICE DESC;
```

Object oriented databases

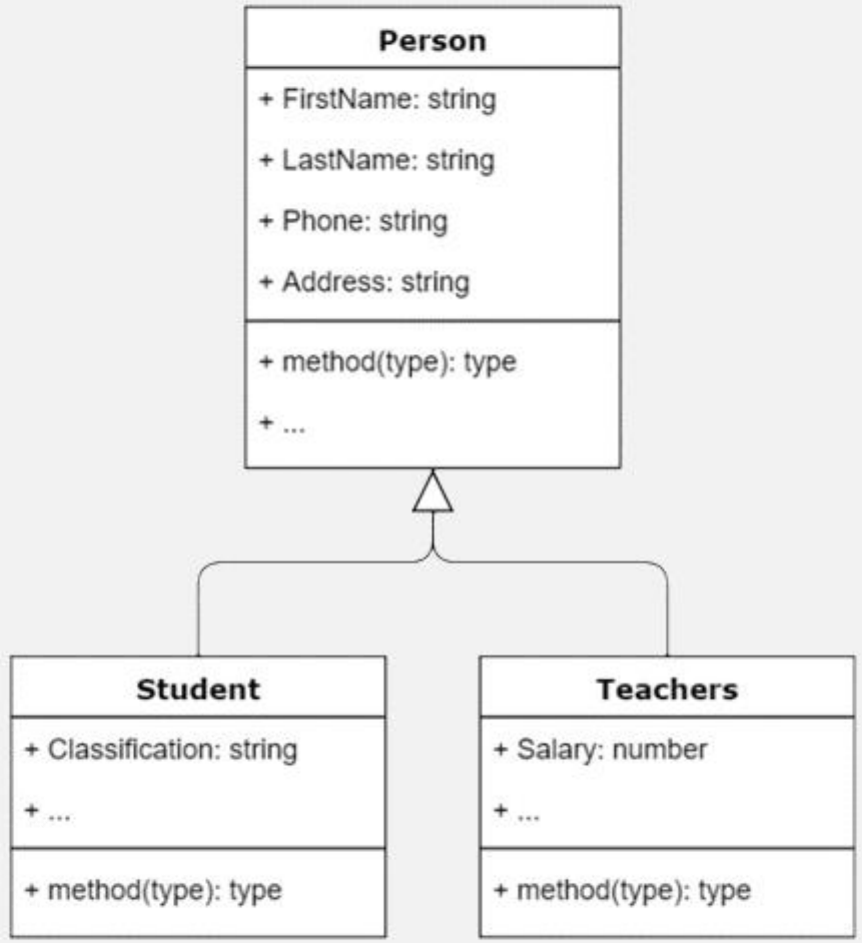
In object-oriented databases, data is stored using object structures, which are defined as classes.

They are not defined by any official standard.

Object oriented databases #2

A characteristic feature of object-oriented databases is that they store objects of arbitrary structures along with the methods attached to them.

This approach gives significant advantage over other types of databases when it comes to storing very complex structures.



Relational-object-oriented databases

Relational-object databases allow you to manipulate data as a set of objects, but have a relational database as the internal storage mechanism.

Streaming databases

Streaming databases are databases in which data are represented as data streams.
Such a database management system is called DSMS (Data Stream Management System).

The data stream model assumes that some or all of the incoming data to the system is not available at any time. The possible time in which they can be recorded is finite. This data appears in the data source and takes the form of a data stream.

Non relational (NoSQL) databases

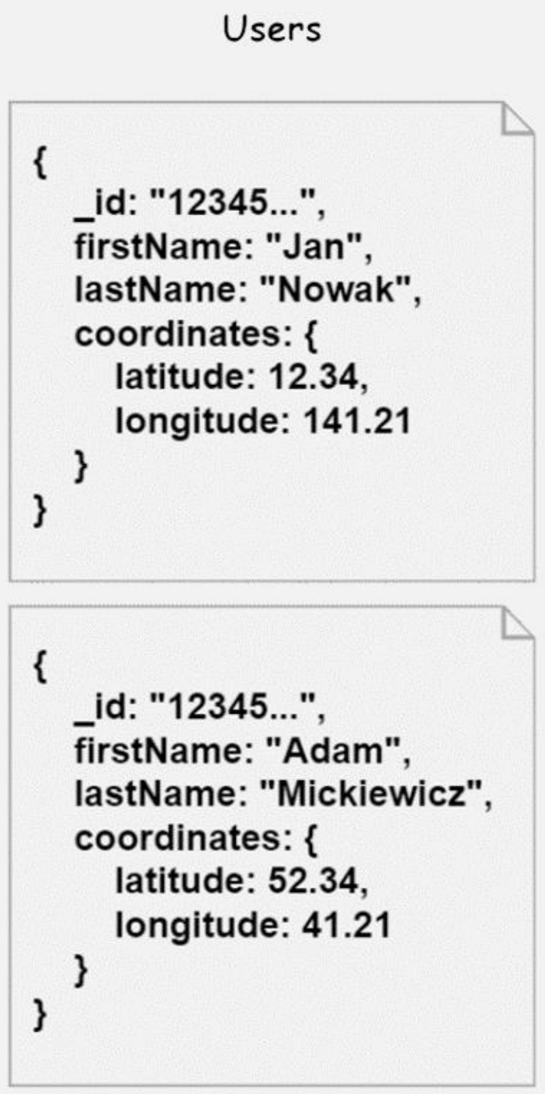
NoSQL (non-relational SQL database) is a database that provides a mechanism for storing and retrieving data modeled in a different way than the tabular relationships used in SQL database relationships.

NoSQL was created out of the need to support larger volumes of data, which forced a shift to a model of building platforms on clusters of less powerful servers.

Non relational databases (NoSQL)

Non relational (NoSQL) database is typically understood to store data as a list of key-value pairs, with no relational relationships between the stored objects.

In a NoSQL database, there is no requirement that the objects be homogeneous in structure.

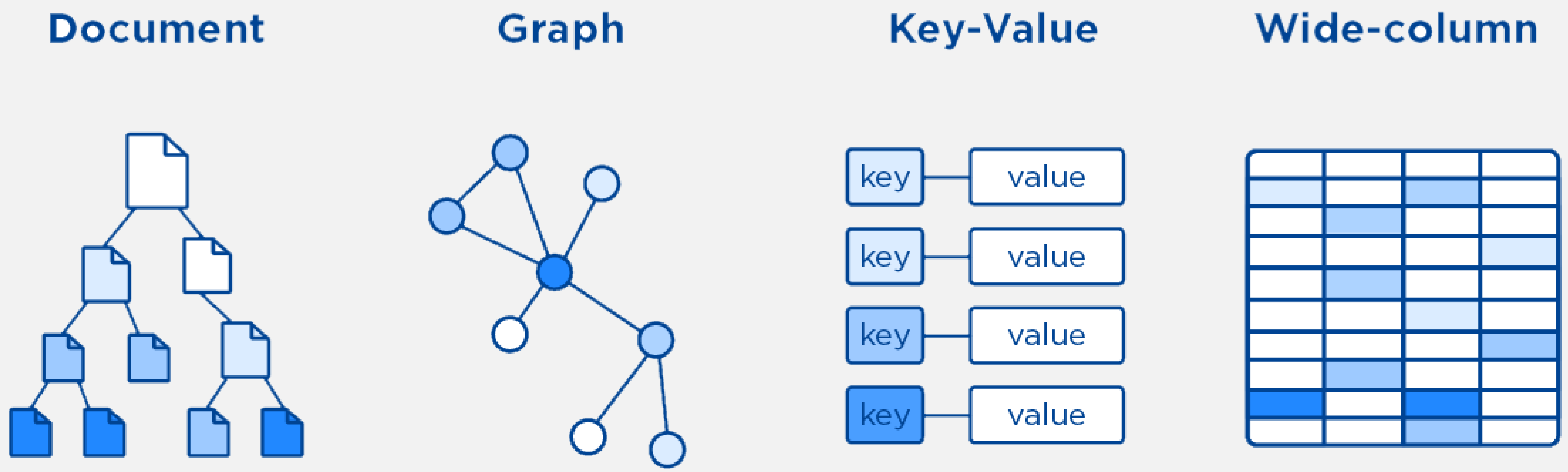


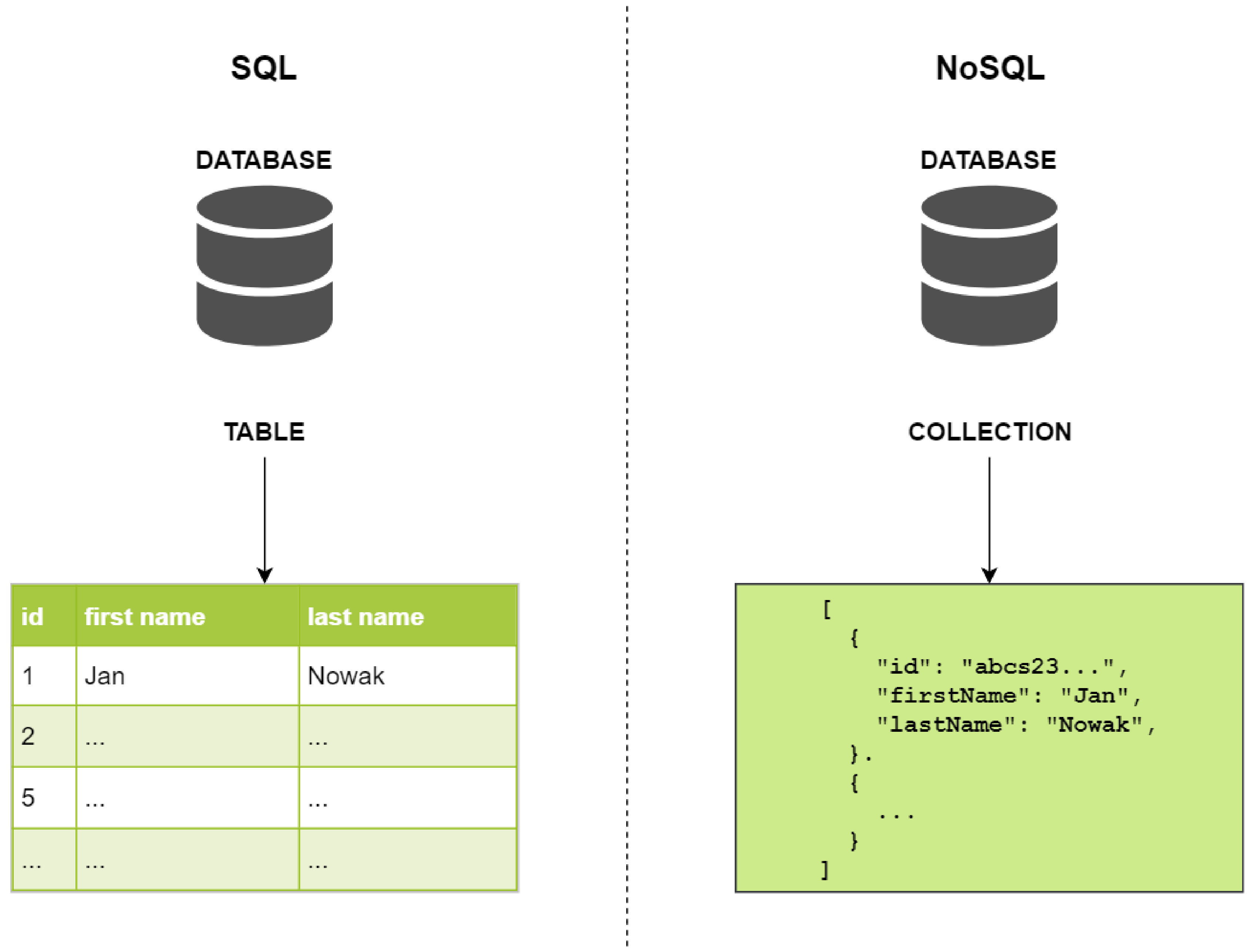
```
{
    _id: "12345...",
    firstName: "Jan",
    lastName: "Nowak",
    coordinates: {
        latitude: 12.34,
        longitude: 141.21
    }
}
```

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

Non relational (NoSQL) databases

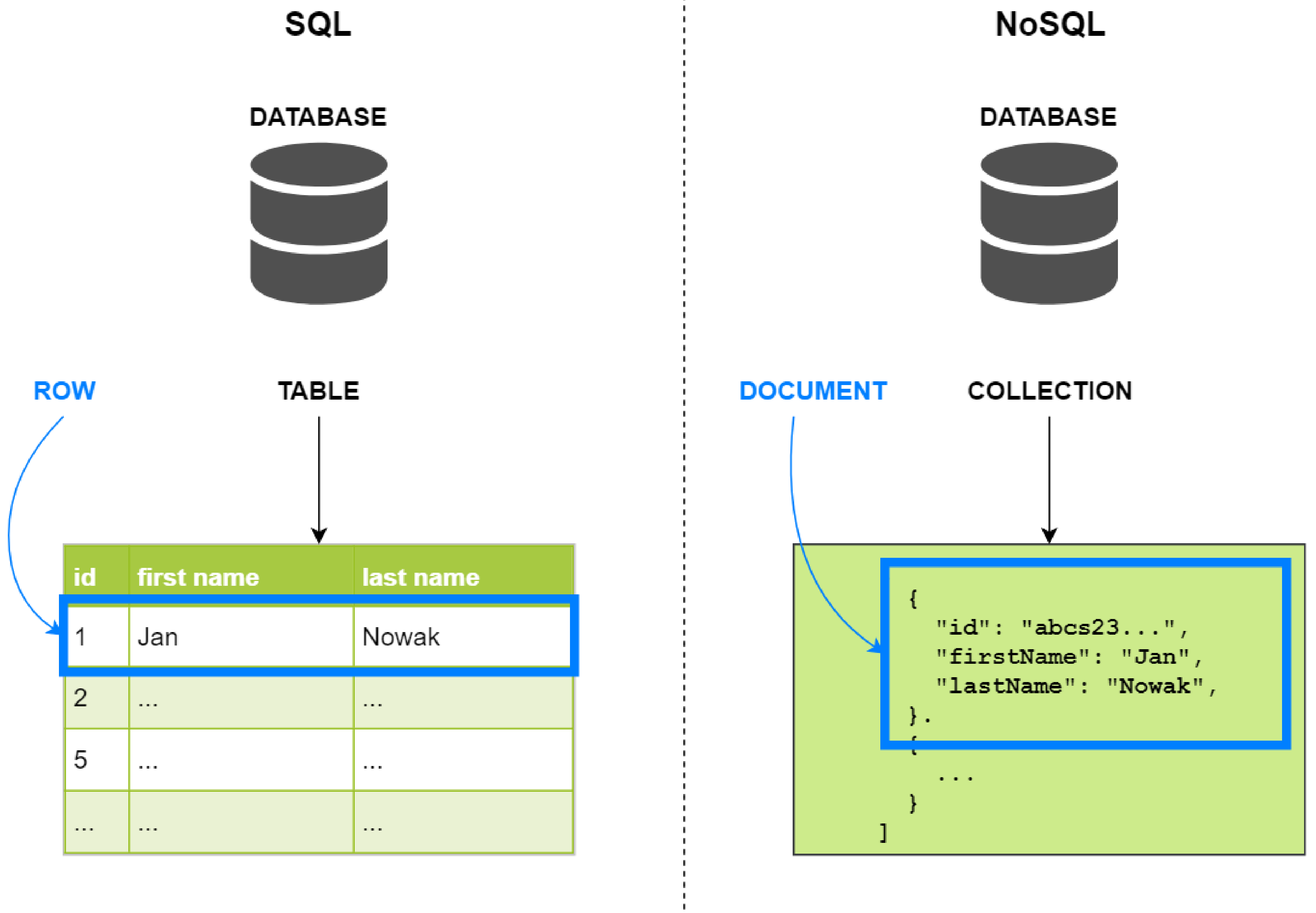
The data structures used by NoSQL (e.g., key-value, graph, document, wide-column) are different from those used by default in relational databases, making some NoSQL operations faster.





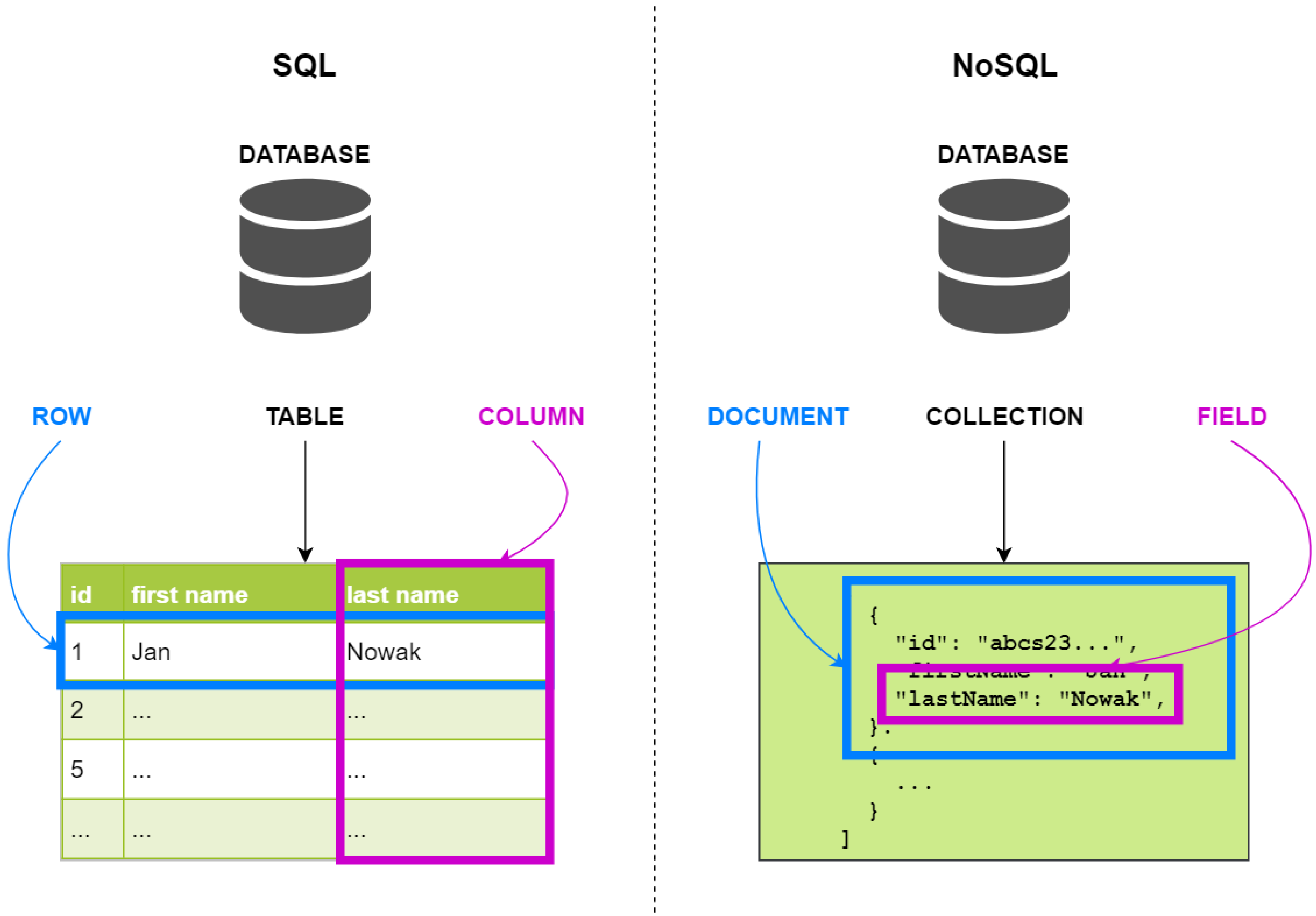
id	first name	last name
1	Jan	Nowak
2	...	...
5	...	...
...	...	...

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



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

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



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

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

MongoDB

MongoDB - an cross-platform, open source, non-relational database management system.
It is characterized by high scalability, performance, and the lack of a strictly defined structure of supported databases.

Data is stored as JSON-style documents, which allows JavaScript applications to process it more naturally, while maintaining the ability to create hierarchies and indexing.

https://www.mongodb.com

Where to Use MongoDB?

- Big Data
- Content Management and Delivery
- Mobile and Social Infrastructure
- User Data Management
- Data Hub

MongoDB - document

Document – is a set of key-value pairs. Documents have dynamic schema – it means that documents in the same collection do not need to have the same set of fields or structure, and common fields in a collection's documents may hold different types of data.

MongoDB - document

```
{
    _id: ObjectId(7df78ad8902c)
    title: 'MongoDB Overview',
    description: 'MongoDB is no sql database',
    tags: ['mongodb', 'database', 'NoSQL'],
    likes: 100,
    comments: [
       {
          user:'user1',
          message: 'My first comment',
          dateCreated: new Date(2031,1,20,2,15),
          like: 0
       }
   ]
 }
```

MongoDB - collection

Collection - group of MongoDB documents.
It is the equivalent of an RDBMS table.
Collections do not enforce a schema.
Documents within a collection can have different fields.
Typically, all documents in a collection are of similar or related purpose.

MongoDB - database

Database - a container for collections. Each database gets its own set of files on the file system.

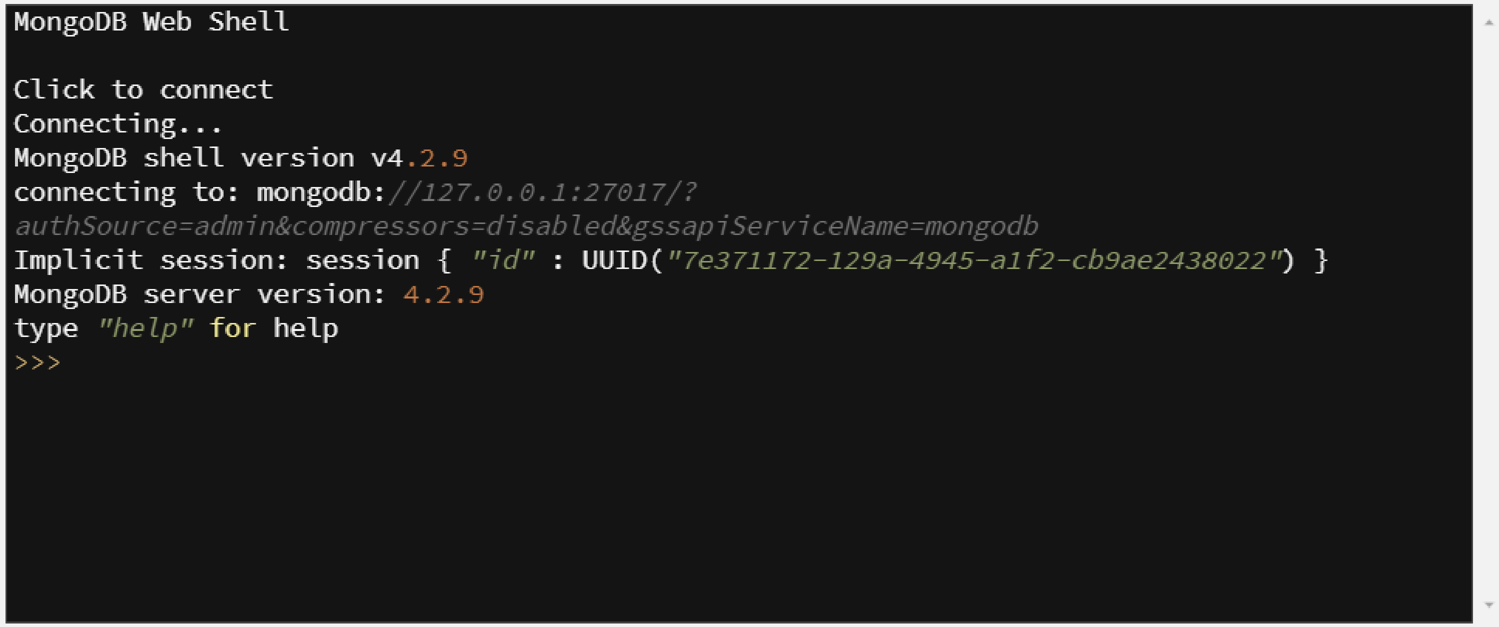
A single MongoDB server typically has multiple databases.

MongoDB shell

The mongo shell is an interactive JavaScript interface to MongoDB.

You can use the mongo shell to query and update data as well as perform administrative operations.

https://docs.mongodb.com/manual/mongo/



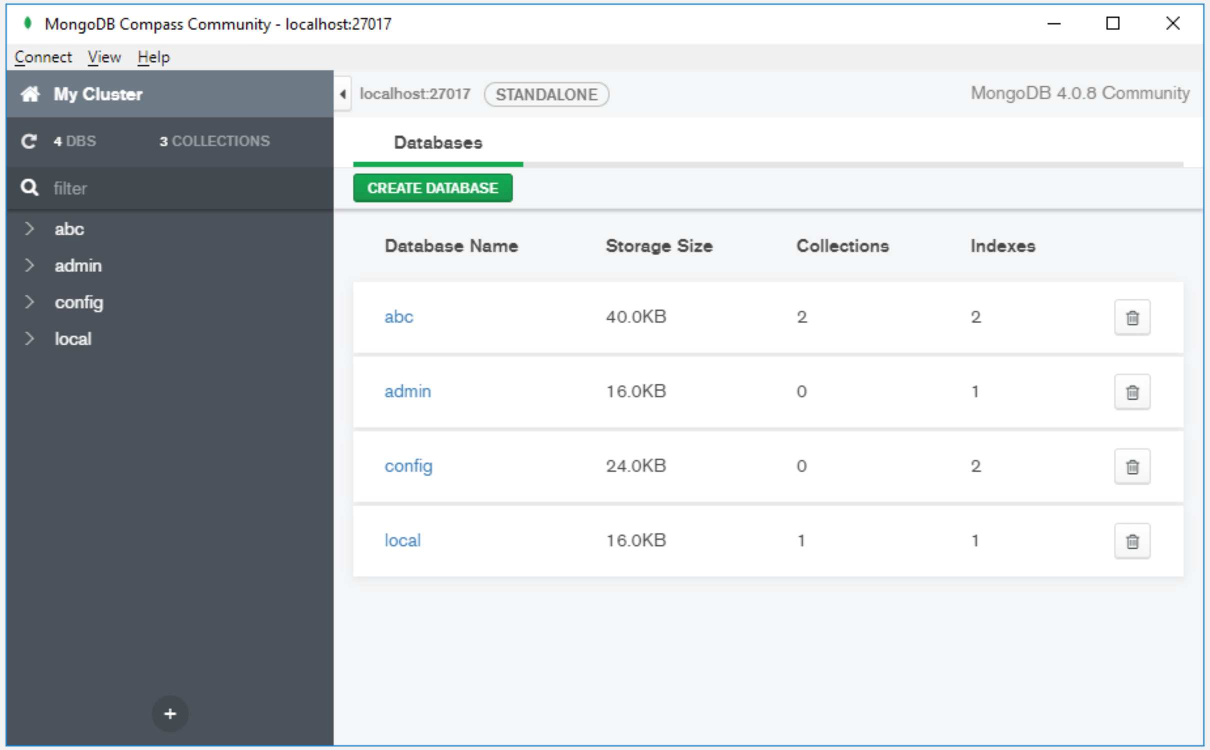
```
MongoDB Web Shell

Click to connect
Connecting...
MongoDB shell version v4.2.9
connecting to: mongodb://127.0.0.1:27017/?
authSource=admin&compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("7e371172-129a-4945-a1f2-cb9ae2438022") }
MongoDB server version: 4.2.9
type "help" for help
>>>
```

MongoDB Compass

GUI for MongoDB, allows you to make smarter decisions about document structure, querying, indexing, document validation, and more.

https://www.mongodb.com/try/download/compass



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

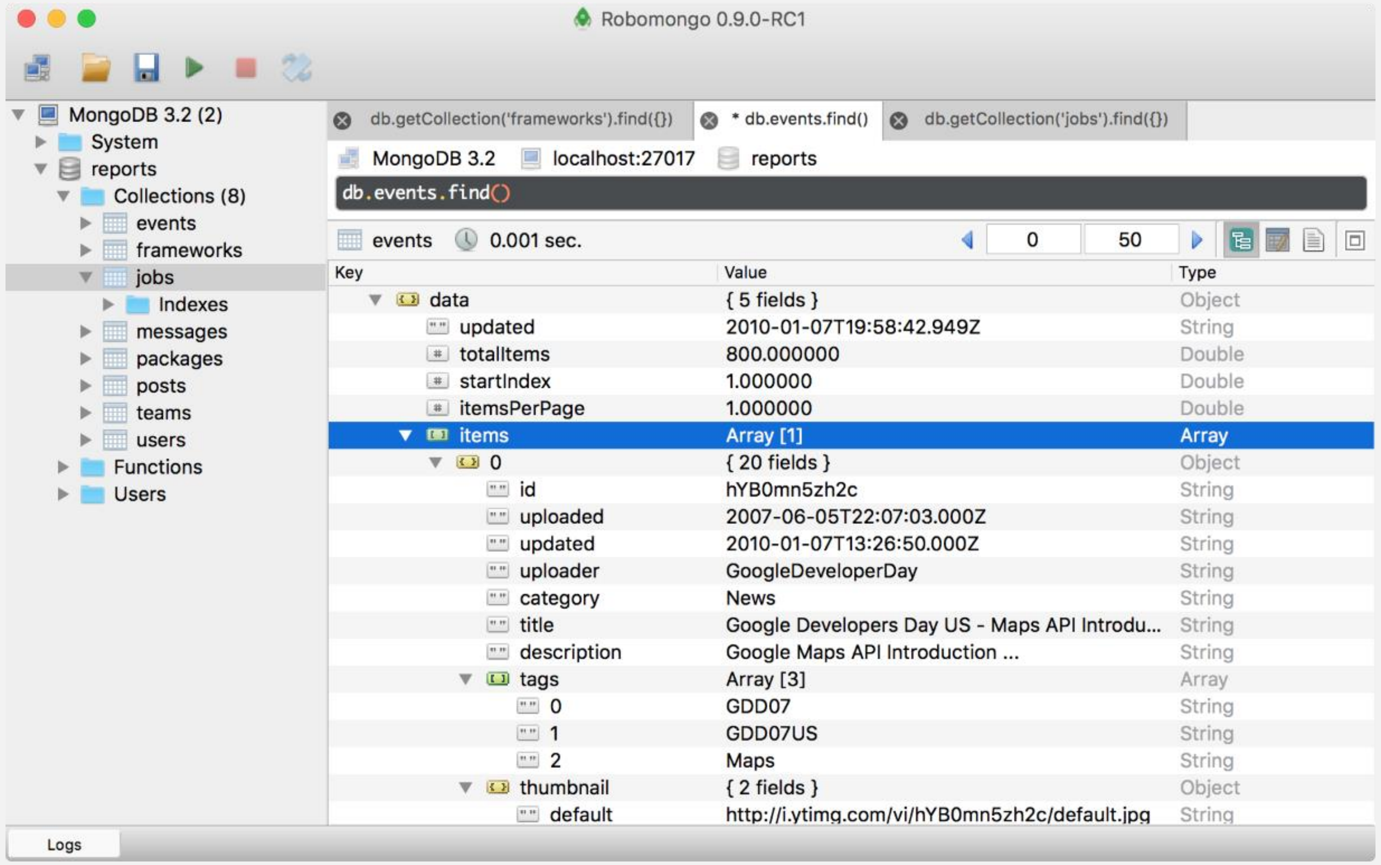
Robo 3T (third party software)

The Robo 3T (formerly Robomongo) is native and cross-platform MongoDB manager.

https://robomongo.org

https://robomongo.org/download

Robo 3T is free, Studio 3T is paid


Robomongo 0.9.0-RC1

- MongoDB 3.2 (2)
  - System
  - reports
    - Collections (8)
      - events
      - frameworks
      - jobs
        - Indexes
      - messages
      - packages
      - posts
      - teams
      - users
    - Functions
    - Users

db.getCollection('frameworks').find({}) | * db.events.find() | db.getCollection('jobs').find({})

MongoDB 3.2 | localhost:27017 | reports

```
db.events.find()
```

events 0.001 sec. 0 50

Key	Value	Type
data	{ 5 fields }	Object
updated	2010-01-07T19:58:42.949Z	String
totalItems	800.000000	Double
startIndex	1.000000	Double
itemsPerPage	1.000000	Double
items	Array [1]	Array
0	{ 20 fields }	Object
id	hYB0mn5zh2c	String
uploaded	2007-06-05T22:07:03.000Z	String
updated	2010-01-07T13:26:50.000Z	String
uploader	GoogleDeveloperDay	String
category	News	String
title	Google Developers Day US - Maps API Introdu...	String
description	Google Maps API Introduction ...	String
tags	Array [3]	Array
0	GDD07	String
1	GDD07US	String
2	Maps	String
thumbnail	{ 2 fields }	Object
default	http://i.ytimg.com/vi/hYB0mn5zh2c/default.jpg	String

Logs

MongoDB Atlas

Global cloud database service for modern applications.
Fully managed MongoDB in cloud with automation and proven practices that guarantee availability, scalability, and compliance with data security and privacy standards.

https://www.mongodb.com/cloud/atlas

MongoDB with NodeJS

The Node.js driver is an interface through which you can connect to and communicate with MongoDB instances.

https://www.npmjs.com/package/mongodb

https://docs.mongodb.com/drivers/node/current/