

# Benchmark baz danych

Aleks Zieliński, Filip Kalinowski

June 18, 2025

## Rozdziały

<b>1</b>	<b>Informacje wstępne</b>	<b>2</b>
1.1	Założenia projektu . . . . .	2
1.2	Użyte technologie i sprzęt . . . . .	2
1.3	Schemat logiczny bazy danych . . . . .	3
1.4	Przykładowe dane z plików JSON . . . . .	3
<b>2</b>	<b>Instalacja baz danych</b>	<b>4</b>
2.1	PostgreSQL . . . . .	4
2.2	MongoDB . . . . .	6
<b>3</b>	<b>Zapytania</b>	<b>8</b>
<b>4</b>	<b>Zapytania w MongoDB</b>	<b>9</b>
4.1	Zapytanie 1 . . . . .	9
4.2	Zapytanie 2 . . . . .	10
4.3	Zapytanie 3 . . . . .	11
4.4	Zapytanie 4 . . . . .	12
4.5	Zapytanie 5 . . . . .	14
4.6	Zapytanie 6 . . . . .	16
4.7	Zapytanie 7 . . . . .	17
4.8	Zapytanie 8 . . . . .	18
<b>5</b>	<b>Zapytania w PostgreSQL</b>	<b>19</b>
<b>6</b>	<b>Zapytania w PostgreSQL w JSONB</b>	<b>19</b>
<b>7</b>	<b>Porównanie czasowe zapytań</b>	<b>19</b>

# 1 Informacje wstępne

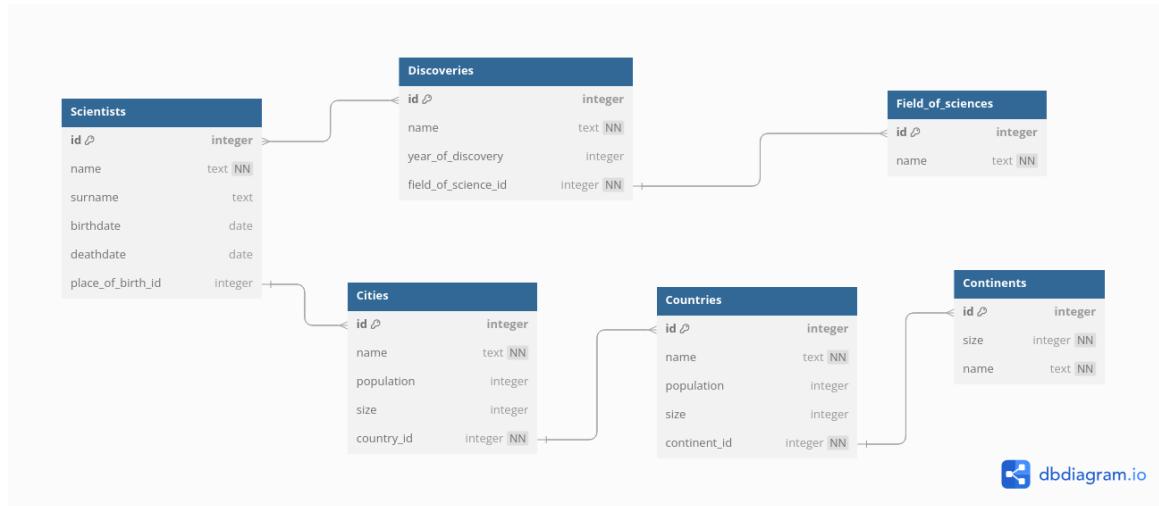
## 1.1 Założenia projektu

- Zaprojektować schemat dokumentów JSON dla wybranego tematu. Należy mieć min 2 typy dokumentów na osobę w projekcie. Minimum jeden typ musi mieć dokumenty zagnieżdżone
- Opracowany model, wraz z propozycją najważniejszych/najbardziej rozbudowanych poleceń należy zatwierdzić u wykładowcy (ew. poprawić wg. sugestii)
- Dla każdego typu dokumentu wygenerować min 100 sensownych instancji. Jeden z typów powinien mieć minimum 500 instancji. Dane można generować automatami typu generatedata.com bądź napisać 4 skrypt w pythonie (dodatkowe punkty).
- Tak przygotowane dokumenty należy zainportować do:
  1. Wybranej bazy NoSQL
  2. PostgreSQLa do kolumn typu JSONB
  3. PostgreSQLa przy jednoczesnej konwersji JSON na tabele
- Wszystkie 3 procesy importowania należy udokumentować screenshotami i zapisanymi wykorzystanymi poleceniami
- Należy utworzyć min 4 zapytania na osobę realizujące najważniejsze problemy wyszukiwania w tworzonej bazie.
- Każde z zapytań musi mieć 3 wersje dla 3 sytuacji z pkt 4.
- Należy przeprowadzić eksperyment pomiaru czasu wykonania tych zapytań. Aby eksperyment był rzetelnie przeprowadzony, zarówno baza NoSQLowa, jak i PostgreSQL muszą operować w podobnych warunkach: albo obie bazy są postawione w osobnych dockerach, albo zainstalowane w tym samym OS, ale uruchamiane jedna na raz (proces instalacji musi być udokumentowany). Wyniki czasowe należy przedstawić w tabelce

## 1.2 Użyte technologie i sprzęt

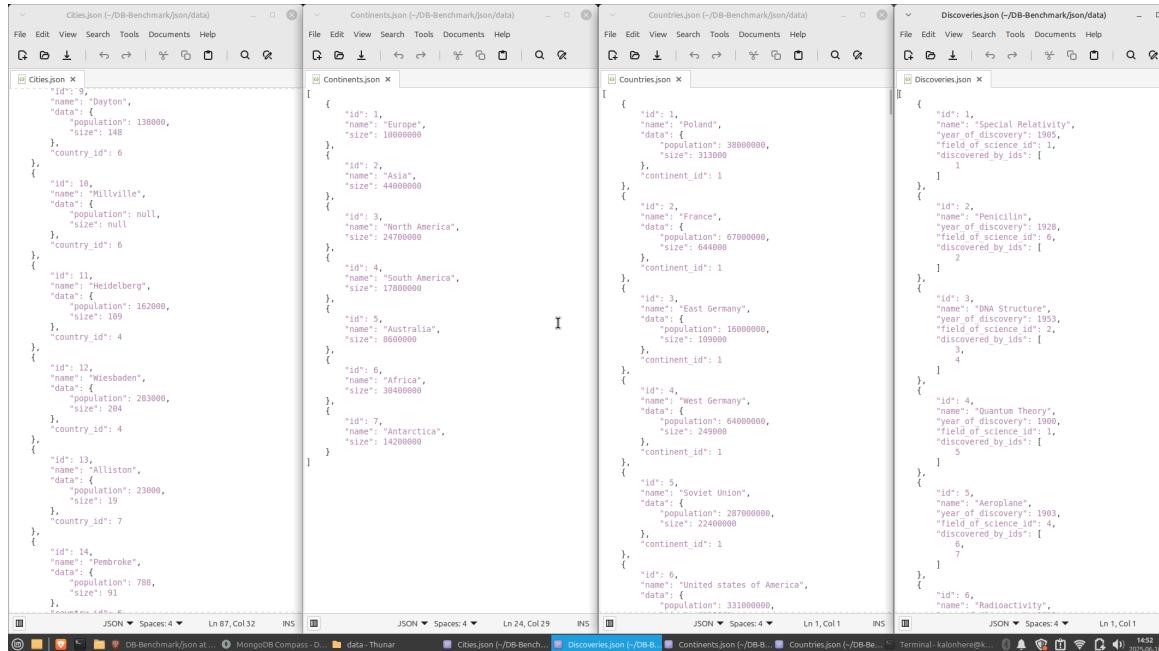
- PostgreSQL - wersja 16.9
- MongoDB - wersja 8.0.10
- Laptop:
  - OS - Linux Mint 22.1
  - CPU - Intel i7-9750H
  - GPU - Nvidia GTX 1660
  - RAM - 16GB 2667 MHz

### 1.3 Schemat logiczny bazy danych



Zdjęcie. 1: Schemat logiczny zrobiony w serwisie dbdiagram.io

### 1.4 Przykładowe dane z plików JSON



Zdjęcie. 2: Pliki JSON od lewej: Cities, Continents, Countries, Discoveries

```

Field_of_sciences.json
[{"id":1,"name":"Physics"}, {"id":2,"name":"Biology"}, {"id":3,"name":"Chemistry"}, {"id":4,"name":"Technology"}, {"id":5,"name":"Medicine"}, {"id":6,"name":"Mathematics"}]

Scientists.json
[{"id": 1, "full_name": { "name": "Albert", "surname": "Einstein" }, "dates": { "birthdate": "1879-03-14", "deathdate": "1955-04-18" }, "place_of_birth_id": 1}, {"id": 2, "full_name": { "name": "Alexander", "surname": "Fleming" }, "dates": { "birthdate": "1881-08-06", "deathdate": "1955-03-11" }, "place_of_birth_id": 2}, {"id": 3, "full_name": { "name": "Francis", "surname": "Crick" }, "dates": { "birthdate": "1916-06-08", "deathdate": "2004-07-28" }, "place_of_birth_id": 3}, {"id": 4, "full_name": { "name": "James", "surname": "Watson" }, "dates": { "birthdate": "1928-04-06", "deathdate": "2024-12-01" }, "place_of_birth_id": 4}, {"id": 5, "full_name": { "name": "Karl", "surname": "Popper" }, "dates": { "birthdate": "1902-07-28", "deathdate": "1994-09-17" }, "place_of_birth_id": 5}]

```

**Zdjęcie. 3:** Pliki JSON od lewej: Field\_of\_sciences, Scientists

## 2 Instalacja baz danych

### 2.1 PostgreSQL

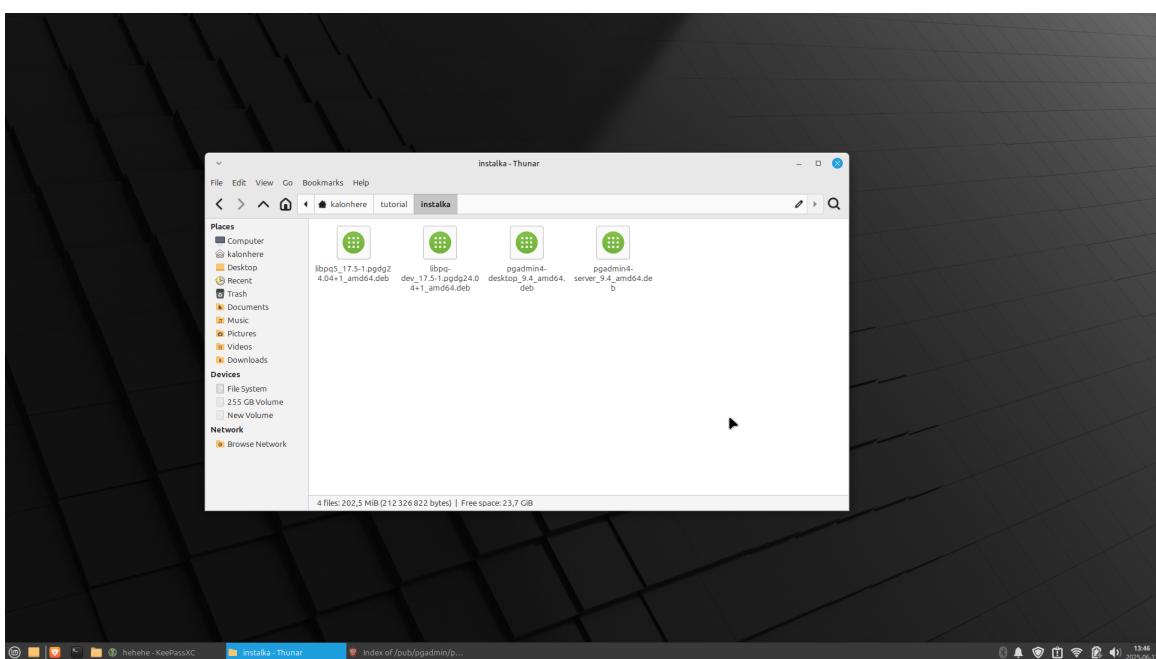
```

Terminal - kalonhere@kalonhere-laptop:~ - 
File Edit View Terminal Tabs Help
kalonhere@kalonhere-laptop:~ $ sudo apt install postgresql postgresql-contrib
$: command not found
kalonhere@kalonhere-laptop:~ $ sudo apt install postgresql postgresql-contrib
[sudo] password for kalonhere: *****

```

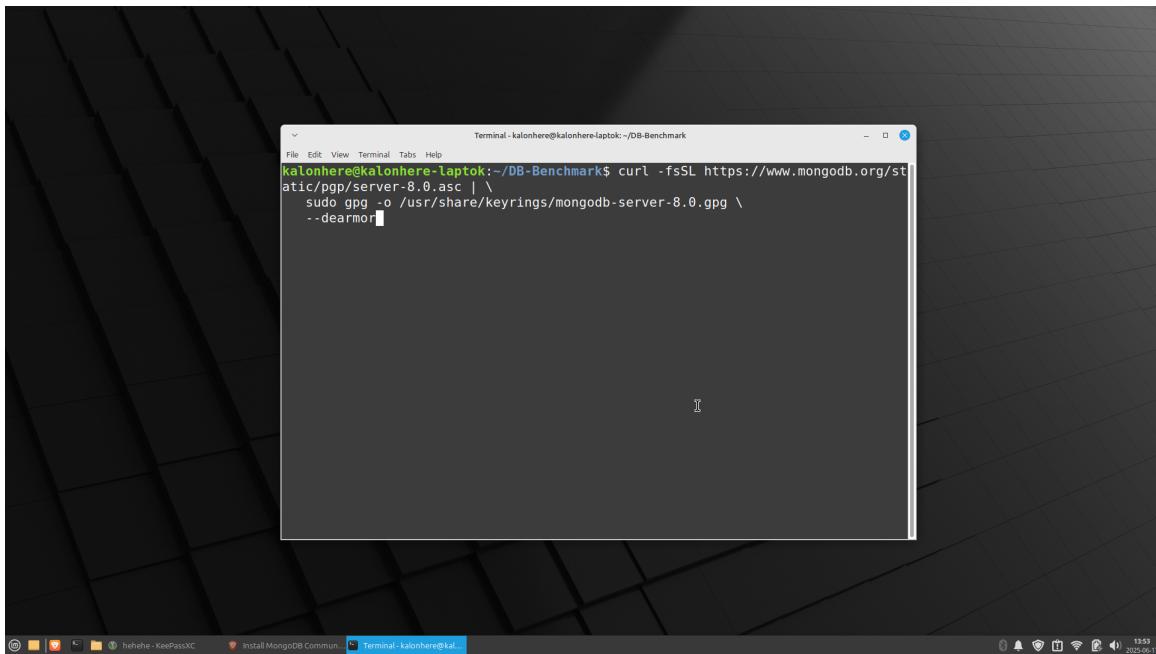
**Zdjęcie. 4:** Instalacja za pomocą apt

**Zdjęcie 5:** Po lewej - komenda systemctl aby sprawdzić czy działa, po prawej - logowanie na user, uruchomienie psql i test działania

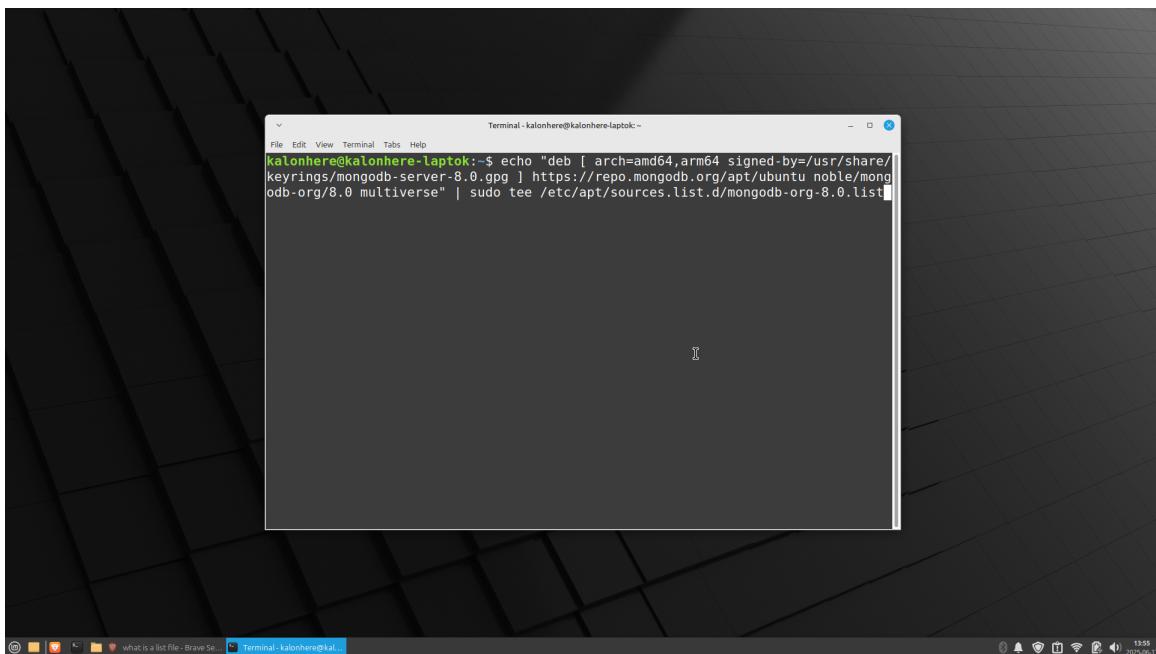


**Zdjęcie. 6:** Instalacja pgadmina z serwera ftp PostgreSQL

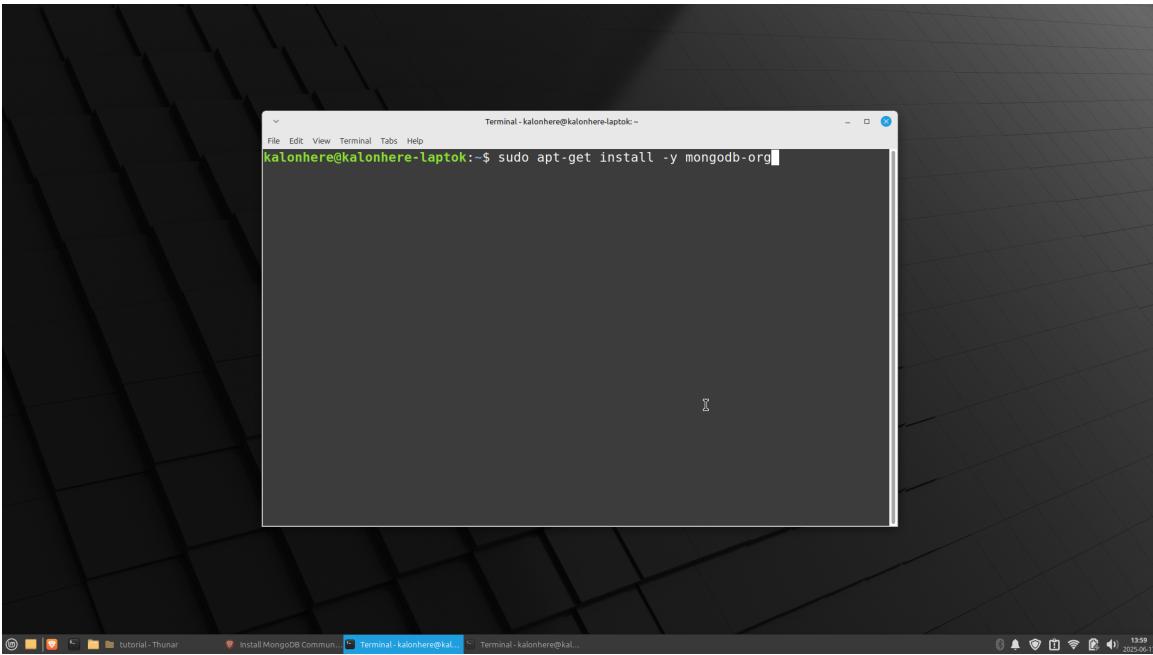
## 2.2 MongoDB



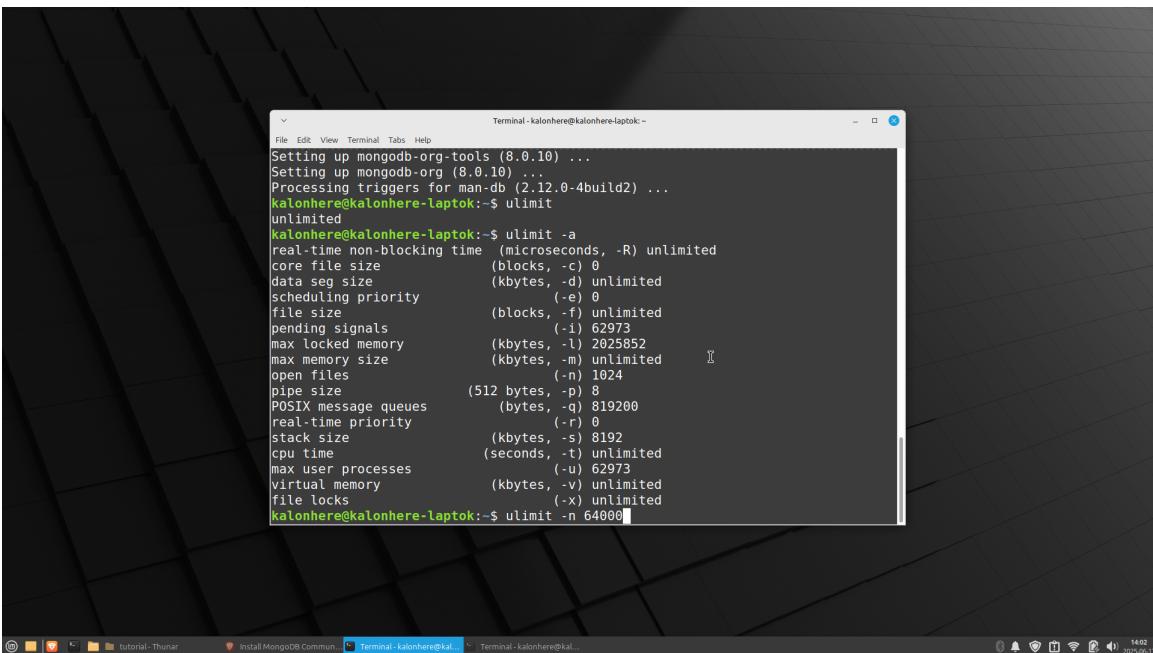
Zdjęcie. 7: Importowanie kluczy gpg do MongoDB



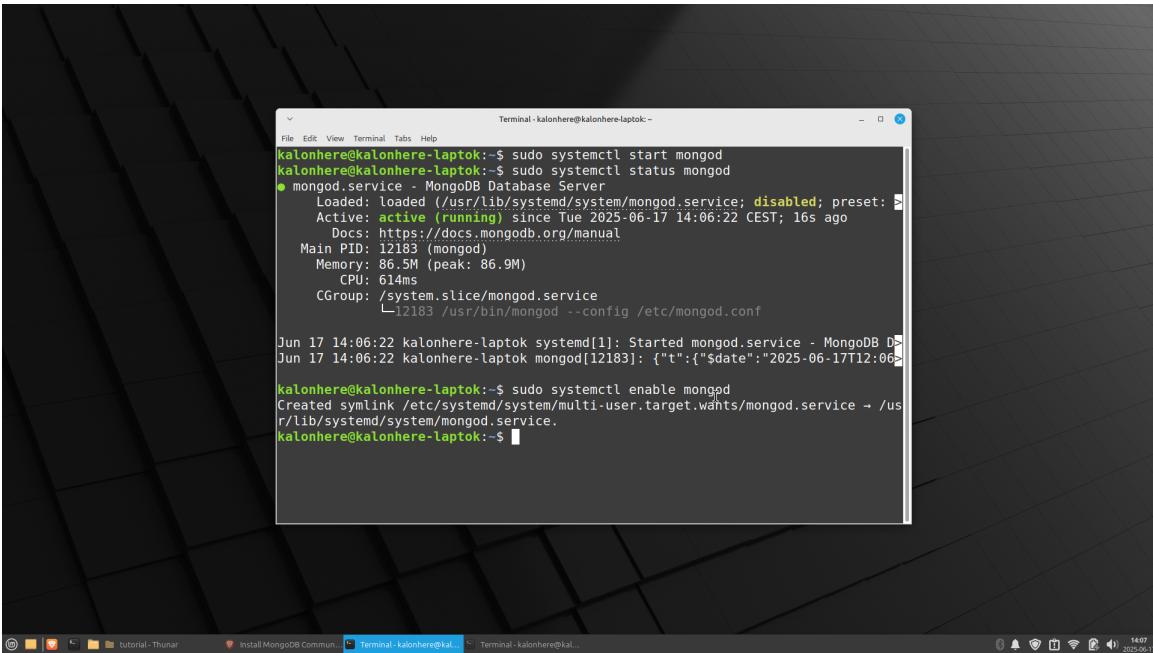
Zdjęcie. 8: Stworzenie pliku z listą źródeł do pobierania repo/package



Zdjęcie. 9: Instalacja MongoDB



Zdjęcie. 10: Zwiększenie limitu otwartych plików z 1024 do 64000



A screenshot of a terminal window titled "Terminal - kalonhere@kalonhere-laptop: ~". The window displays the following command-line session:

```
File Edit View Terminal Tabs Help
kalonhere@kalonhere-laptop:~$ sudo systemctl start mongod
kalonhere@kalonhere-laptop:~$ sudo systemctl status mongod
● mongod.service - MongoDB Database Server
   Loaded: loaded (/usr/lib/systemd/system/mongod.service; disabled; preset: )
   Active: active (running) since Tue 2025-06-17 14:06:22 CEST; 16s ago
     Docs: https://docs.mongodb.org/manual
   Main PID: 12183 (mongod)
      Memory: 86.5M (peak: 86.9M)
        CPU: 614ms
      CGroup: /system.slice/mongod.service
              └─12183 /usr/bin/mongod --config /etc/mongod.conf

Jun 17 14:06:22 kalonhere-laptop[1]: Started mongod.service - MongoDB D
Jun 17 14:06:22 kalonhere-laptop[1]: {"t": {"$date": "2025-06-17T12:06:22.000+02:00"}, "c": "main", "m": "mongod (pid 12183) starting up.", "v": 2}
kalonhere@kalonhere-laptop:~$ sudo systemctl enable mongod
Created symlink /etc/systemd/system/multi-user.target.wants/mongod.service → /usr/lib/systemd/system/mongod.service.
kalonhere@kalonhere-laptop:~$
```

Zdjęcie. 11: Uruchomienie MongoDB, sprawdzenie statusu i włączenie autouruchamiania

### 3 Zapytania

1. Wybierz państwo którego nazwa kończy się na "a" i ma najwięcej odkryć z fizyki
2. Znajdź najstarsze i najmłodsze odkrycie z USA z biologii
3. Znajdź państwa które mają odkrycia z biologii po 1960 roku
4. Znajdź średnią liczbę wszystkich odkryć na państwo w każdym kontynencie
5. Wypisz dziedzine w której każde państwo z europy dokonało najwięcej odkryć
6. Wypisz średnią liczbę odkryć na naukowca z każdego kontynentu
7. Znajdź państwo z największą średnią liczbą odkryć na kilometr kwadratowy
8. Znajdź miasto z którego pochodzi najwięcej fizyków

## 4 Zapytania w MongoDB

### 4.1 Zapytanie 1

```
[{$match: {
    name: { $regex: "a$", $options: "i" }
},
{$lookup: {
    from: "Cities",
    localField: "id",
    foreignField: "country_id",
    as: "cities"
}
},
{ $unwind: "$cities" },
{$lookup: {
    from: "Scientists",
    localField: "cities.id",
    foreignField: "place_of_birth_id",
    as: "scientists"
}
},
{ $unwind: "$scientists" },
{$lookup: {
    from: "Discoveries",
    let: { scientistId: "$scientists.id" },
    pipeline: [
        {
            $match: {
                $expr: {
                    $and: [
                        { $eq: ["$field_of_science_id", 1] },
                        { $in: ["$$scientistId", "$discovered_by_ids"] }
                    ]
                }
            }
        ],
        as: "discoveries"
    }
},
{ $unwind: "$discoveries" },
{$group: {
    _id: { country_id: "$id", country: "$name", discovery_id: "$discoveries.id", discovery_name: "$discoveries.name" }
},
{$group: {
    _id: { country_id: "$_id.country_id", country: "$_id.country" },
    discoveries: { $addToSet: { id: "$_id.discovery_id", name: "$_id.discovery_name" } },
    count: { $sum: 1 }
}
},
{ $sort: { count: -1 } },
{ $limit: 1 },
{$project: {
    _id: 0,
    country: "$_id.country",
    count: 1,
    discoveries: 1
}
}]]
```

## 4.2 Zapytanie 2

```
[{$match: {
    field_of_science_id: 2
},
{$unwind: "$discovered_by_ids"
},
{$lookup: {
    from: "Scientists",
    localField: "discovered_by_ids",
    foreignField: "id",
    as: "scientist"
}
},
{$unwind: "$scientist" },
{$lookup: {
    from: "Cities",
    localField: "scientist.place_of_birth_id",
    foreignField: "id",
    as: "birth_city"
}
},
{$unwind: "$birth_city" },
{$match: {
    "birth_city.country_id": 6
}
},
{$group: {
    _id: {
        id: "$id",
        name: "$name",
        year_of_discovery: "$year_of_discovery"
    }
}
},
{$sort: { "_id.year_of_discovery": 1 } },
{$group: {
    _id: null,
    discoveries: { $push: "$_id" }
}
},
{$project: {
    _id: 0,
    oldest: { $arrayElemAt: ["$discoveries", 0] },
    youngest: { $arrayElemAt: ["$discoveries", -1] }
}
}]]
```

### 4.3 Zapytanie 3

```
[{$match: {
    field_of_science_id: 2,
    year_of_discovery: { $gt: 1960 }
},
{ $unwind: "$discovered_by_ids",
{$lookup: {
    from: "Scientists",
    localField: "discovered_by_ids",
    foreignField: "id",
    as: "scientist"
}
},
{ $unwind: "$scientist" },
{$lookup: {
    from: "Cities",
    localField: "scientist.place_of_birth_id",
    foreignField: "id",
    as: "city"
}
},
{ $unwind: "$city" },
{$group: {
    _id: {
        country_id: "$city.country_id",
        discovery_id: "$id",
        discovery_name: "$name"
    }
}
},
{$lookup: {
    from: "Countries",
    localField: "_id.country_id",
    foreignField: "id",
    as: "country"
}
},
{ $unwind: "$country" },
{$group: {
    _id: "$country.name",
    discoveries: { $addToSet: { id: "$_id.discovery_id", name: "$_id.discovery_name" } },
    count: { $sum: 1 }
}
},
{$sort: { count: -1 } },
{$project: {
    _id: 0,
    country: "$_id",
    count: 1,
    discoveries: 1
}
}]]
```

#### 4.4 Zapytanie 4

```
[{$lookup: {
  from: "Scientists",
  localField: "discovered_by_ids",
  foreignField: "id",
  as: "scientist"
}
},
{ $unwind: "$scientist" },
{$lookup: {
  from: "Cities",
  localField: "scientist.place_of_birth_id",
  foreignField: "id",
  as: "city"
}
},
{ $unwind: "$city" },
{$lookup: {
  from: "Countries",
  localField: "city.country_id",
  foreignField: "id",
  as: "country"
}
},
{ $unwind: "$country" },
{$lookup: {
  from: "Continents",
  localField: "country.continent_id",
  foreignField: "id",
  as: "continent"
}
},
{ $unwind: "$continent" },
{$group: {
  _id: {
    continent_id: "$continent.id",
    continent_name: "$continent.name",
    country_id: "$country.id",
    country_name: "$country.name",
    discovery_id: "$id"
  }
}
},
{$group: {
  _id: {
    continent_id: "$_id.continent_id",
    continent_name: "$_id.continent_name",
    country_id: "$_id.country_id",
    country_name: "$_id.country_name" },
    discoveries_count: { $sum: 1 }
  }
}]
```

```
},
{$group: {
  _id: {
    continent_id: "$_id.continent_id",
    continent_name: "$_id.continent_name"},
    average_discoveries_per_country: { $avg: "$discoveries_count" },
    countries: {
      $push: {
        country_id: "$_id.country_id",
        country_name: "$_id.country_name",
        discoveries_count: "$discoveries_count"
      }
    }
  }
},
{
  $project: {
    _id: 0,
    continent_id: "$_id.continent_id",
    continent_name: "$_id.continent_name",
    average_discoveries_per_country: { $round: ["$average_discoveries_per_country", 2] },
  }
},
{$sort: { continent_name: 1 } }
]
```

## 4.5 Zapytanie 5

```
[{$unwind: "$discovered_by_ids"
},
{$lookup: {
    from: "Scientists",
    localField: "discovered_by_ids",
    foreignField: "id",
    as: "scientist"
}
},
{ $unwind: "$scientist" },
{$lookup: {
    from: "Cities",
    localField: "scientist.place_of_birth_id",
    foreignField: "id",
    as: "city"
}
},
{ $unwind: "$city" },
{$lookup: {
    from: "Countries",
    localField: "city.country_id",
    foreignField: "id",
    as: "country"
}
},
{ $unwind: "$country" },
{$match: {
    "country.continent_id": 1
}
},
{$group: {
    _id: {
        country_id: "$country.id",
        country_name: "$country.name",
        field_of_science_id: "$field_of_science_id"
    },
    discoveries_count: { $sum: 1 }
}
},
{$sort: {
    "_id.country_name": 1,
    discoveries_count: -1
}
},
{$group: {
    _id: "$_id.country_name",
    field_of_science_id: { $first: "$_id.field_of_science_id" },
    discoveries_count: { $first: "$discoveries_count" }
}
},
```

```
    {$lookup: {
        from: "Field_of_sciences",
        localField: "field_of_science_id",
        foreignField: "id",
        as: "science"
    },
    { $unwind: "$science" },
    {$project: {
        _id: 0,
        country: "$_id",
        field_of_science: "$science.name",
        discoveries_count: 1
    }
},
{$sort: { country: 1 } }
```

## 4.6 Zapytanie 6

```
[{$lookup: {
    from: "Cities",
    localField: "place_of_birth_id",
    foreignField: "id",
    as: "city"
},
{$unwind: "$city" },
{$lookup: {
    from: "Countries",
    localField: "city.country_id",
    foreignField: "id",
    as: "country"
},
{$unwind: "$country" },
{$lookup: {
    from: "Continents",
    localField: "country.continent_id",
    foreignField: "id",
    as: "continent"
},
{$unwind: "$continent" },
{$lookup: {
    from: "Discoveries",
    let: { scientistId: "$id" },
    pipeline: [
        {
            $match: {
                $expr: { $in: ["$$scientistId", "$discovered_by_ids"] }
            }
        ],
        as: "discoveries"
    }
},
{$addFields: {
    discoveries_count: { $size: "$discoveries" }
}},
{$group: {
    _id: "$continent.name",
    total_discoveries: { $sum: "$discoveries_count" },
    scientists_count: { $sum: 1 }
}},
{$project: {
    _id: 0,
    continent: "$_id",
    avg_discoveries_per_scientist: {
        $cond: [
            { $eq: ["$scientists_count", 0] },
            0,
            { $round: [{ $divide: ["$total_discoveries", "$scientists_count"] }, 2] }
        ]
    },
    total_discoveries: 1,
    scientists_count: 1
}},
{$sort: { continent: 1 } }
]]
```

## 4.7 Zapytanie 7

```
[{$unwind: "$discovered_by_ids"
},
{$lookup: {
    from: "Scientists",
    localField: "discovered_by_ids",
    foreignField: "id",
    as: "scientist"
}
},
{$unwind: "$scientist" },
{$lookup: {
    from: "Cities",
    localField: "scientist.place_of_birth_id",
    foreignField: "id",
    as: "city"
}
},
{$unwind: "$city" },
{$lookup: {
    from: "Countries",
    localField: "city.country_id",
    foreignField: "id",
    as: "country"
}
},
{$unwind: "$country" },
{$group: {
    _id: {
        country_id: "$country.id",
        country_name: "$country.name",
        discovery_id: "$id"
    },
    country_size: { $first: "$country.data.size" }
}
},
{$group: {
    _id: {
        country_id: "$_id.country_id",
        country_name: "$_id.country_name"
    },
    discoveries_count: { $sum: 1 },
    country_size: { $first: "$country_size" }
}
},
{$addFields: {
    avg_discoveries_per_km2: {
        $cond: [
            { $gt: ["$country_size", 0] },
            { $divide: ["$discoveries_count", "$country_size"] },
            null
        ]
    }
},
{$sort: { avg_discoveries_per_km2: -1 } },
{$limit: 1 },
{$project: {
    _id: 0,
    country: "$_id.country_name",
    discoveries_count: 1,
    size: "$country_size",
    avg_discoveries_per_km2: { $round: ["$avg_discoveries_per_km2", 6] }
}
}]]
```

## 4.8 Zapytanie 8

```
[{$lookup: {
    from: "Discoveries",
    let: { scientistId: "$id" },
    pipeline: [
        {
            $match: {
                field_of_science_id: 1,
                $expr: { $in: ["$$scientistId", "$discovered_by_ids"] }
            }
        }
    ],
    as: "physics_discoveries"
},
{$match: {
    "physics_discoveries.0": { $exists: true }
},
{$group: {
    _id: "$place_of_birth_id",
    physicists_count: { $sum: 1 }
},
{$lookup: {
    from: "Cities",
    localField: "_id",
    foreignField: "id",
    as: "city"
},
{$unwind: "$city" },
{$sort: { physicists_count: -1 } },
{$limit: 1 },
{$project: {
    _id: 0,
    city: "$city.name",
    country_id: "$city.country_id",
    physicists_count: 1
}
}]]
```

## 5 Zapytania w PostgreSQL

mnkln

## 6 Zapytania w PostgreSQL w JSONB

faafefe

## 7 Porównanie czasowe zapytań

Po wykonaniu każdego zapytania odpowiednio, otrzymaliśmy następujące wyniki:

Zapytanie	MongoDB	PostgreSQL-JSONB	PostgreSQL
1	22ms		
2	4ms		
3	1ms		
4	41ms		
5	49ms		
6	54ms		
7	70ms		
8	42ms		