

MongoDB

Wprowadzenie do baz danych

MongoDB

- Nierelacyjny system zarządzania bazą danych
- Zorientowany na przechowywanie dokumentów JSON
- Nadaje się do przechowywania dużej ilości danych o małej wartości np. komentarzy w mediach społecznościowych, wyników pomiarów, ...
- Dobre współpracuje z aplikacjami używającymi danych w formacie JSON
- Brak ścisłego typowania, obsługa dużej ilości typów danych, nie ma ścisłej struktury dokumentów
- Umożliwia agregację danych
- Pierwsze wydanie 2009
- Aktualne (9 grudnia 2024) 8.0.4

Siedziba firmy MongoDB Inc. w Palo Alto



[Biura Oracle](#)

[Oracle Wa-wa](#)



PayPal

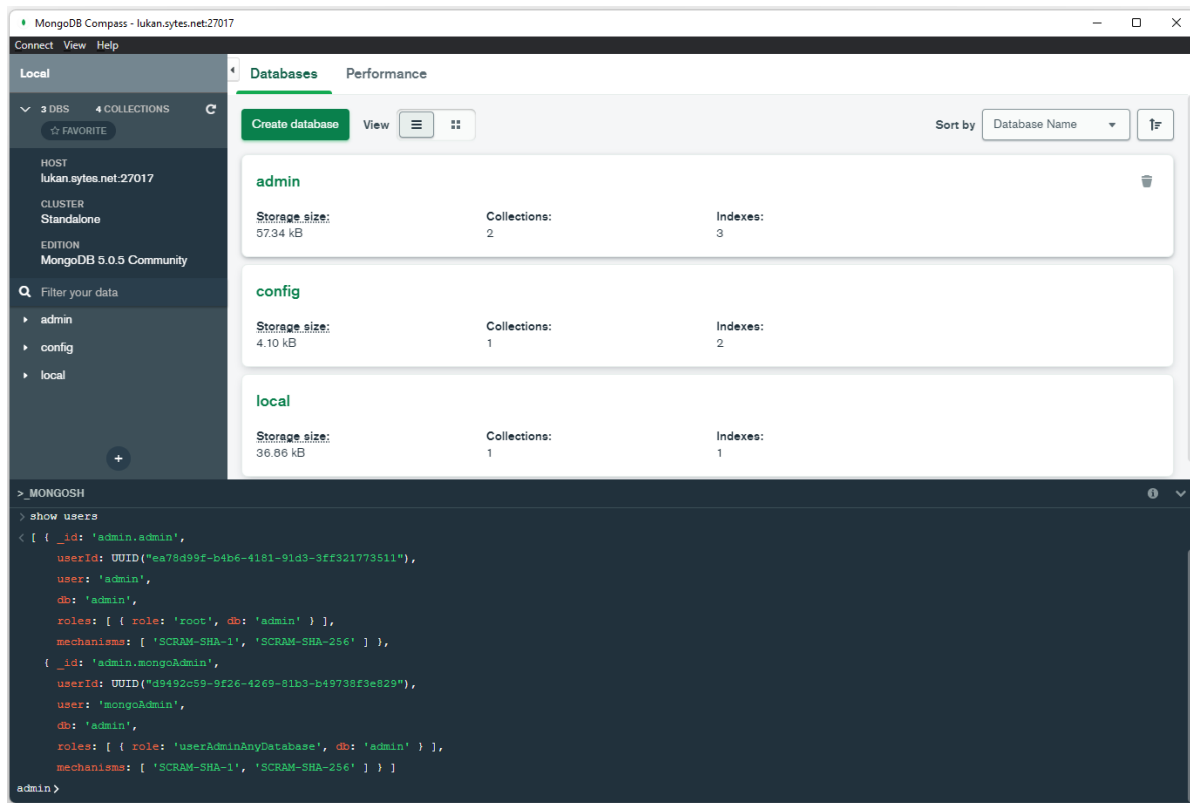


Bezpieczeństwo

Domyślnie MongoDB jest dostępne bez hasła i może pojawić się coś takiego

```
{"_id":{"$_oid":"6054054e6e6a3ac5ca93831c"},"content":"All your data is a backed up. You must pay 0.015 BTC to 16xHLwFYgGFJeMCv3wQ6FFdMUnj7yoQJ46 48 hours for recover it. After 48 hours expiration we will leaked and exposed all your data. In case of refusal to pay, we will contact the General Data Protection Regulation, GDPR and notify them that you store user data in an open form and is not safe. Under the rules of the law, you face a heavy fine or arrest and your base dump will be dropped from our server! You can buy bitcoin here, does not take much time to buy https://localbitcoins.com with this guide https://localbitcoins.com/guides/how-to-buy-bitcoins After paying write to me in the mail with your DB IP: allmydataback@cock.li and you will receive a link to download your database dump."}
```

MongoDB Compass



Wprowadzenie zabezpieczeń

security:
authorization: enabled

plik: mongodb.config

```
use admin
db.createUser(
  { user: "username",
    pwd: "password", roles: [ "root" ]
  }
)
```

```
db.createUser(
{
  user: "mongoAdmin",
  pwd: "changeMe",
  roles: [ { role: "userAdminAnyDatabase", db: "admin" } ]
}
)
```

```
use mydatabase
db.createUser(
{
  user: "username",
  pwd: "password",
  roles: [ { role: "readWrite", db: "mydatabase" } ]
}
)
```

Lista ról:

<https://www.mongodb.com/docs/manual/reference/built-in-roles/>

Terminologia

MongoDB	SQL
database	database
collection	table
document	wiersz – record (row)
pole – field	kolumna – column
linking/embedded documents	join
primary key (_id field, ObjectId)	primary key (user designated)
index	index

```
_id: ObjectId("609079e524a99c04e23aa784")
✓ Nagłówek: Array
  > 0: Object
    ✓ Pozycje faktury: Array
      > 0: Object
        idpoz: 1
        nrfaktury: 1
        nazwa: "Listwa łaz. Ozdobna 10cm"
        idproduktu: "P13 "
        ilosc: 3
        cena_jednostkowa: 16.2
        vat_jednostkowy: 0.07
        rabat_jednostkowy: 0
      > 1: Object
      > 2: Object
    > Suma faktury: Array
```

```
_id: ObjectId("609079e324a99c04e23aa766")
✓ Nagłówek: Array
  > 0: Object
    nazwa: "Posadzki Remonty"
    nip: "99-88732"
    adres: "Różana 13"
    miasto: "Gdańsk"
    kod: "70-231"
    rabat: 0.02
    nrfaktury: 2
    datasp: "2019-01-05T00:00:00"
    zaplaco: "nie"
  > Pozycje faktury: Array
    > 0: Object
      idpoz: 3
      nrfaktury: 2
      nazwa: "Ścisk stolarski"
      idproduktu: "P09 "
      ilosc: 1
      cena_jednostkowa: 99
      vat_jednostkowy: 0.22
      rabat_jednostkowy: 0.02
    > 1: Object
      idpoz: 4
      nrfaktury: 2
      nazwa: "Kątownik 5 cm"
      idproduktu: "P35 "
      ilosc: 3
      cena_jednostkowa: 10
```

Identyfikator obiektu - dokumentu

- Każdy dokument posiada unikalny 12 bajtowy identyfikator `_id`
- Konstrukcja identyfikatora
 - a 4-byte *timestamp value*, representing the ObjectId's creation, measured in seconds since the Unix epoch
 - a 5-byte *random value* (3 bajtowy identyfikator komputera + 2 bajtowy identyfikator procesu)
 - a 3-byte *incrementing counter*, initialized to a random value

```
_id: ObjectId("609079e524a99c04e23aa784")  
✓ Nagłówek: Array  
  > 0: Object  
    > Pozycje faktury: Array  
    > Suma faktury: Array
```

```
_id: ObjectId("609079e324a99c04e23aa766")  
✓ Nagłówek: Array  
  > Pozycje faktury: Array  
  > Suma faktury: Array
```

JavaScript Object Notation (JSON)

```
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021"
  },
  "phoneNumber": [
    { "type": "home",
      "number": "212 555-1234" },
    { "type": "fax",
      "number": "646 555-4567" }
  ],
  "gender": {
    "type": "male"
  }
}
```

```
{
  "_id": {
    "oid": "60970dbf24a99c04e23adfce"
  },
  "Nagłówek": {
    "nazwa": "Budrex Sp z.o.o.",
    "nip": "222-3333",
    "adres": "Kwiatowa 333",
    "miasto": "Gdynia",
    "kod": "80-983",
    "rabat": 0.02,
    "nr faktury": 56,
    "datasp": "2020-05-19T00:00:00",
    "zaplacono": "nie"
  },
  "Pozycje faktury": [
    {
      "idpoz": 158,
      "nr faktury": 56,
      "nazwa": "Cekol drobnodziarnisty",
      "id produktu": "P29 ",
      "ilosc": 1,
      "cena_jednostkowa": 25.3,
      "vat_jednostkowy": 0.07,
      "rabat_jednostkowy": 0.03
    },
    {
      "idpoz": 160,
      "nr faktury": 56,
      "nazwa": "Lakier do drewna LA",
      "id produktu": "P20 ",
      "ilosc": 11,
      "cena_jednostkowa": 23,
      "vat_jednostkowy": 0.22,
      "rabat_jednostkowy": 0.03
    },
    {
      "idpoz": 161,
      "nr faktury": 56,
      "nazwa": "Gładź gipsowa",
      "id produktu": "P30 ",
      "ilosc": 8,
      "cena_jednostkowa": 42.6,
      "vat_jednostkowy": 0.07,
      "rabat_jednostkowy": 0.03
    }
  ],
  "suma faktury": 679.38
}
```

Obiekt: { nazwa : wartość, nazwa : wartość, ... }

Tablica: [obiekt1, obiekt2, ...]

Podstawowe operacje (funkcje) MongoDB

- INSERT (Create)
 - `db.<collectionName>.insert( { .... } )`
- SELECT (Read)
 - `db.<collectionName>.find(query, projection)`
 - `db.<collectionName>.findOne(query, projection)`
 - `db.<collectionName>.aggregate( [ { <stage> }, ... ] )`
- UPDATE
 - `db.<collectionName>.update( { .... } )`
- DELETE
 - `db.<collectionName>.remove( { .... } )`

db.<collectionName>.find(query, projection)

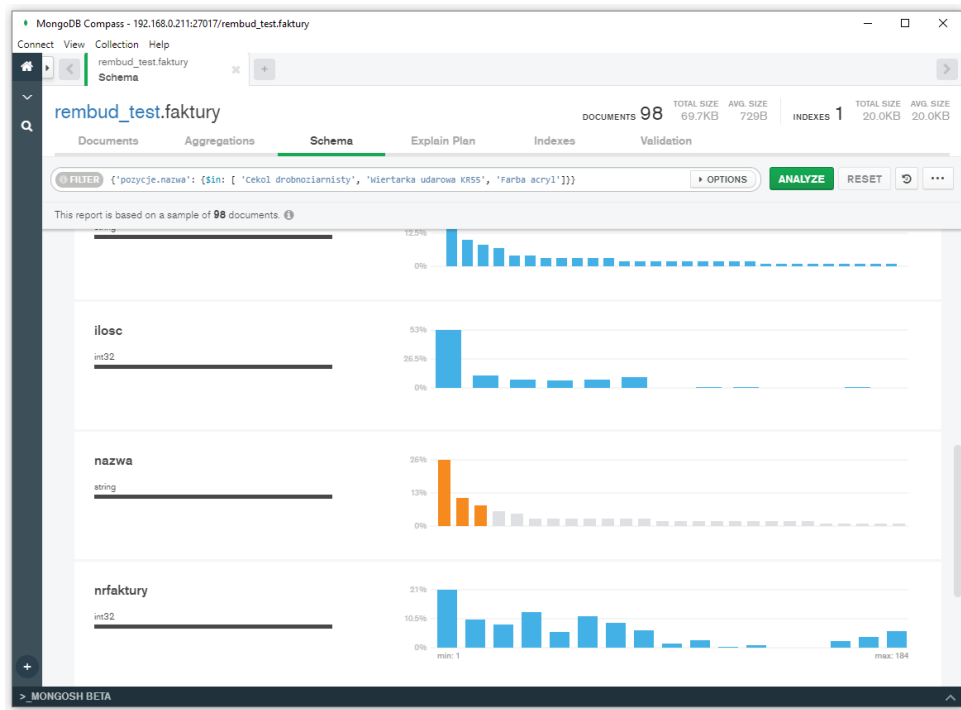
- **query** – reprezentuje opcjonalny warunek selekcji dokumentów (odpowiednik klauzuli WHERE w języku SQL), do kolekcji wyjściowej przechodzą dokumenty spełniające warunek
- **projection** – opcjonalnie określa pola jakie mają znaleźć się w dokumentach wyjściowych, odpowiednik klauzuli SELECT w języku SQL
- db.<collectionName>.find(query, projection).**limit(n)** – ogranicza wynik do n dokumentów
- db.<collectionName>.find(query, projection).**skip(n)** – pomija pierwszych n dokumentów
- db.<collectionName>.find(query, projection).**sort({ field: value })** – sortuje wynik według pola field, value = 1 rosnąco, value = -1 malejąco

Przykład: **db.faktury2.find().sort({"Nagłówek.nrfaktury" : 1}).skip(10).limit(2)**

query

- { <field1>: <value1>, ... }
 - {'Nagłówek.nip': '45-95959'}
 - odpowiednik w języku SQL (SELECT * FROM faktury WHERE "Nagłówek.nip" = '45-95959')
 - {'Nagłówek.miasto': 'Gdynia', 'Nagłówek.zaplacono': 'nie', 'Pozycje faktury.nazwa': 'Wiertarka udarowa KR55'}
- { <field1>: { <operator1>: <value1> }, ... }
 - {'Nagłówek.nip': {\$in: ['45-95959', '76-45940', '89-85858', '99-99999', '99-88732']}}
 - {'Suma faktury': {\$gte: 800, \$lt: 1000}}
 - {'Suma faktury': {\$gte: 800, \$lt: 1000}, 'Nagłówek.zaplacono': 'nie'}

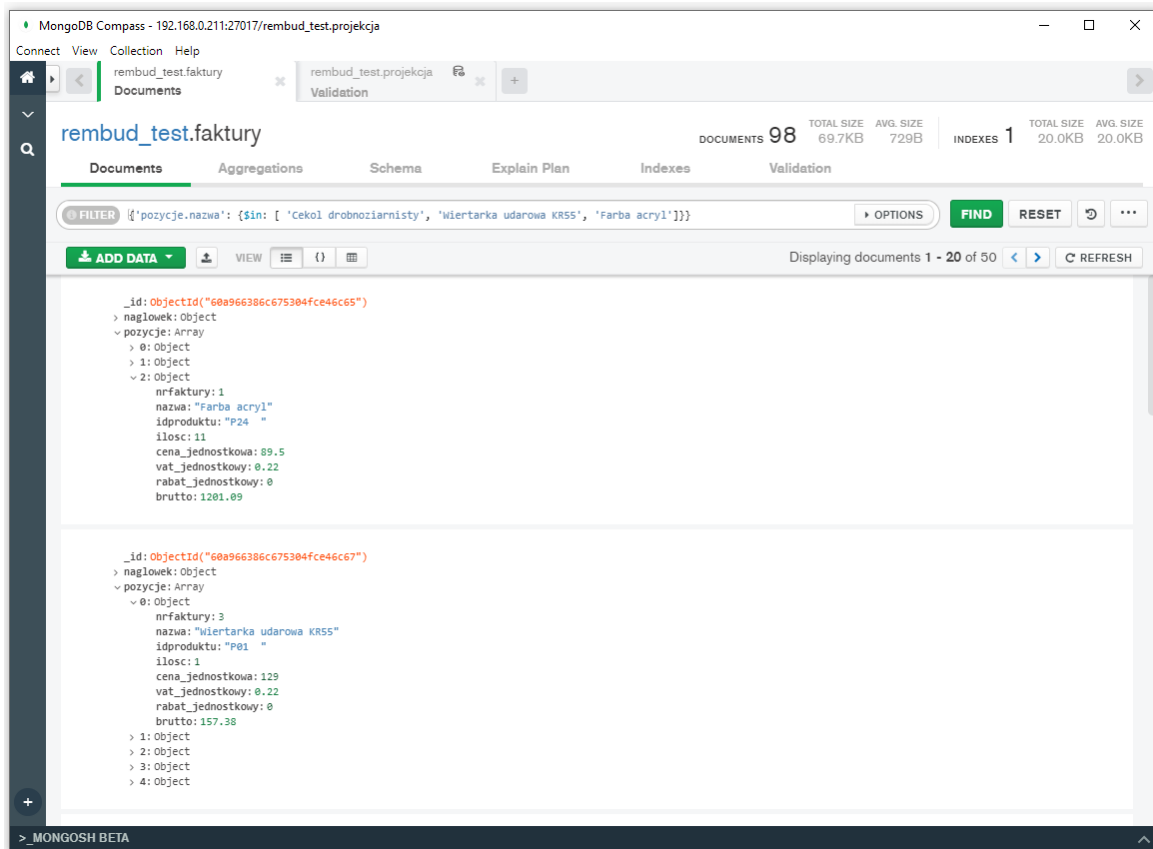
Analiza schematu danych dokumentu w programie Compass



Analizator schematu

1. dokonuje analizy danych bieżącej kolekcji
2. ułatwia tworzenie zapytań

Wynik selekcji w programie Compass



MongoDB Compass - 192.168.0.211:27017/rembud_test.projekcja

Connect View Collection Help

rembud_test.faktury Documents Validation

rembud_test.faktury

DOCUMENTS 98 TOTAL SIZE 69.7KB AVG. SIZE 729B INDEXES 1 TOTAL SIZE 20.0KB AVG. SIZE 20.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER: {'pozycje.nazwa': {'\$in': ['cekol drobnodziarnisty', 'wierarka udarowa KR55', 'Farba acryl']}}

ADD DATA VIEW

Displaying documents 1 - 20 of 50

```
{
  "_id": ObjectId("60a966386c675304fce46c65"),
  "naglowek": Object,
  "pozycje": Array
    0: Object
    1: Object
    2: Object
      nr faktury: 1
      nazwa: "Farba acryl"
      id produktu: "P24"
      ilosc: 11
      cena jednostkowa: 89.5
      vat_jednostkowy: 0.22
      rabat_jednostkowy: 0
      brutto: 1201.09
  }
}
```

```
{
  "_id": ObjectId("60a966386c675304fce46c67"),
  "naglowek": Object,
  "pozycje": Array
    0: Object
      nr faktury: 3
      nazwa: "wierarka udarowa KR55"
      id produktu: "P01"
      ilosc: 1
      cena jednostkowa: 129
      vat_jednostkowy: 0.22
      rabat_jednostkowy: 0
      brutto: 157.38
    1: Object
    2: Object
    3: Object
    4: Object
  }
}
```

> _MONGOSH BETA

W kolekcji wyjściowej są tylko faktury, w których figurują wskazane towary

Operatory porównań i operatory logiczne

Oznaczenie	Opis
\$eq	Sprawdza czy dwie wartości są równe
\$gt	Sprawdza czy pierwsza wartość jest większa od drugiej
\$gte	Sprawdza czy pierwsza wartość jest większa równa od drugiej
\$lt	Sprawdza czy pierwsza wartość jest mniejsza od drugiej
\$lte	Sprawdza czy pierwsza wartość jest mniejsza równa od drugiej
\$ne	Sprawdza czy pierwsza wartość jest różna od drugiej
\$in	Sprawdza czy którykolwiek z podanych argumentów należy do zbioru
\$nin	Sprawdza czy żaden z podanych argumentów nie należy do zbioru
\$or	Logiczne OR
\$and	Logiczne AND
\$not	Logiczna negacja
\$nor	Logiczne NOR
\$exists	Sprawdza czy dane pole jest w dokumencie

Operatory porównań i operatory logiczne

Oznaczenie	Opis
\$eq	{"time.Autor" : {\$eq : "Piotr"}}
\$gt	{"pomiar.x" : {\$gt : 99}}
\$gte	Sprawdza czy pierwsza wartość jest większa równa od drugiej
\$lt	{"Suma faktury": {\$lt : 300}}
\$lte	Sprawdza czy pierwsza wartość jest mniejsza równa od drugiej
\$ne	Sprawdza czy pierwsza wartość jest różna od drugiej
\$in	{"miernik":{\$in : ["Testowy", "Nowy"]}}
\$nin	Sprawdza czy żaden z podanych argumentów nie należy do zbioru
\$or	Logiczne OR
\$and	{\$and : [{"payload.x" : {\$gt : 90.0}}, {"payload.x" : {\$lt : 99.0}}]}
\$not	Logiczna negacja
\$nor	Logiczne NOR
\$exists	{"pomiar.x" : {\$exists : true}}, {"x":{\$exists : true}}, {"miernik":{\$exists : true}}

Projekcja - projection

Format { <field1>: <value>, <field2>: <value> ... }

Projection	Description
<field>: <1 or true>	Dodaje pole do kolekcji wyjściowej
<field>: <0 or false>	Usuwa pole do kolekcji wyjściowej
<field>: <aggregation expression>	Dodaje pole podsumowania do kolekcji wyjściowej (umożliwia zmianę nazwy pola)

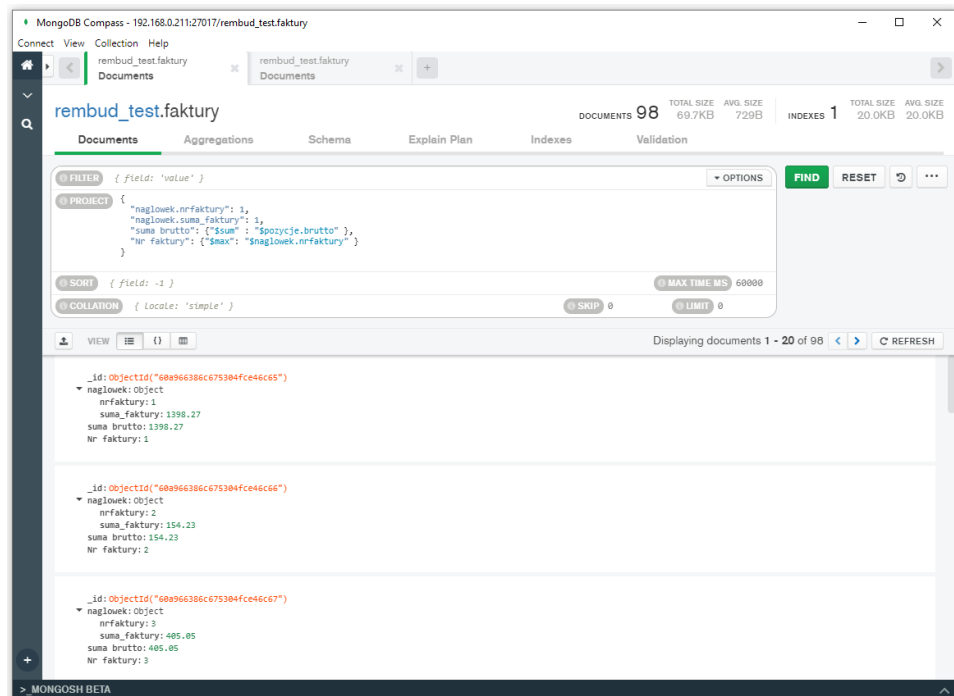
```
{
  "Nagłówek.nrfaktury": 1,
  "Suma faktury": 1,
  "Łączna ilość": {"$sum": "$Pozycje faktury.ilosc"},
  "Nr faktury": {"$max": "$naglowek.nrfaktury"}
}
```

```
{
  "naglowek.nrfaktury": 1,
  "naglowek.suma_faktury": 1,
  "suma brutto": {"$sum": "$pozycje.brutto" },
  "Nr faktury": {"$max": "$naglowek.nrfaktury" }
}
```

FAKTURY.faktury2

Projekcja dostępna w funkcji aggregate() daje większe możliwości formowania dokumentu wyjściowego

Wynik projekcji w MongoDB Compass



```
{
  "naglowek.nrfaktury": 1,
  "naglowek.suma_faktury": 1,
  "suma brutto": { "$sum": "$pozycje.brutto" },
  "Nr faktury": { "$max": "$naglowek.nrfaktury" }
}
```

Funkcje agregujące (akumulujące) dane

{ <operator>: [<argument1>, <argument2> ...] }

{ <operator>: <argument> }

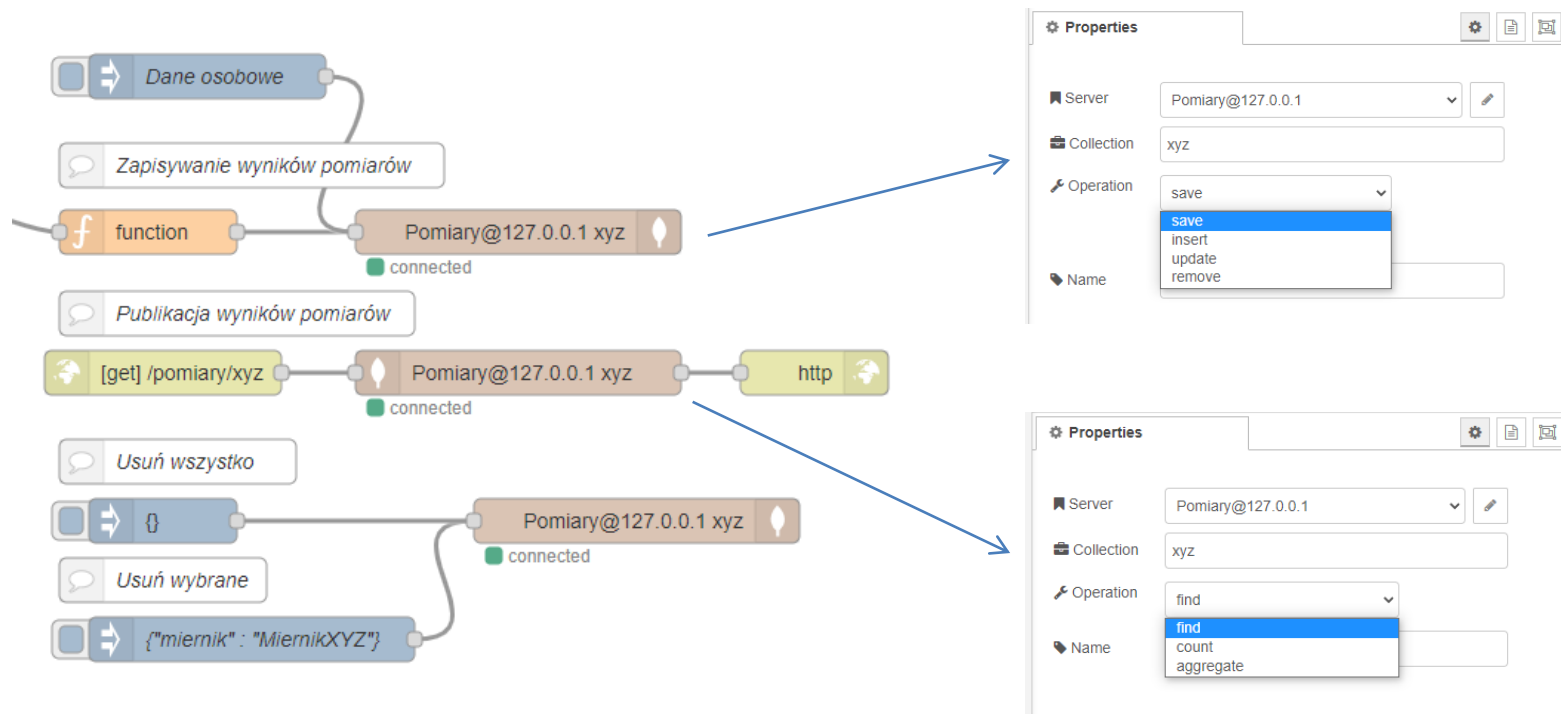
Funkcja (operator)	Użycie
\$sum	{ \$sum: <expression> }, { \$sum: [<expression1>, <expression2> ...] } {"\$sum" : "\$pozycje.brutto" }, {"\$sum" : [3, 3, 4] }
\$max	
\$min	
\$avg	
\$first	
\$stdDevPop	

Wszystkie funkcje w tym matematyczne

$+$, $-$, $\times$, $/$

<https://docs.mongodb.com/manual/reference/operator/aggregation/>

Zapisywanie i usuwanie danych – Pomiar

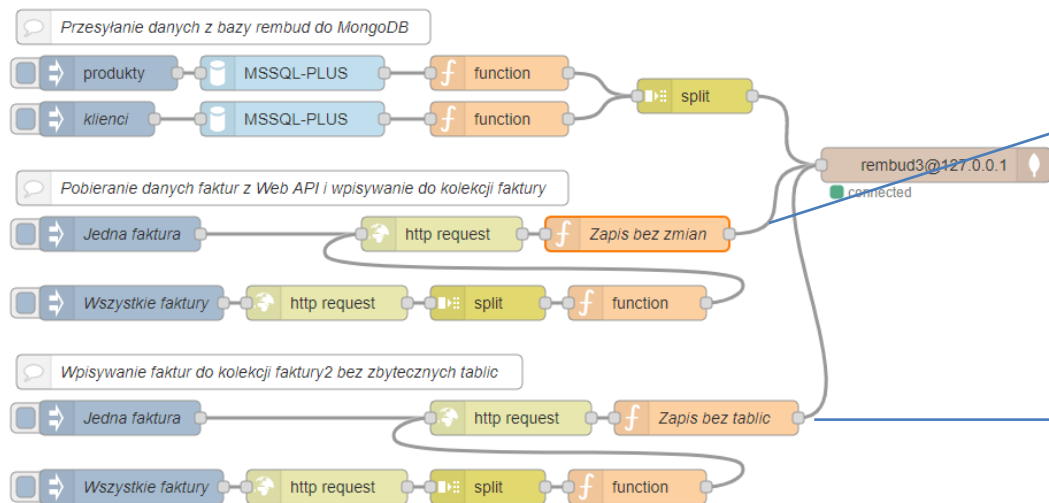


Zakładka: TI MQTT pulpit sterownia

<https://lukan.sytes.net:1881/pomiar/xyz>

Przeniesienie danych z RemBud z serweraSQL do MongoDB

1. Przeniesienie tabel produkty i klienci do MongoDB
2. Pobieranie danych faktur z usługi REST Web API <http://argo.am.gdynia.pl/www/APIRembud/> i zapisanie ich w kolekcji faktury i faktury 2



Name: Zapis bez zmian

Setup Function Close

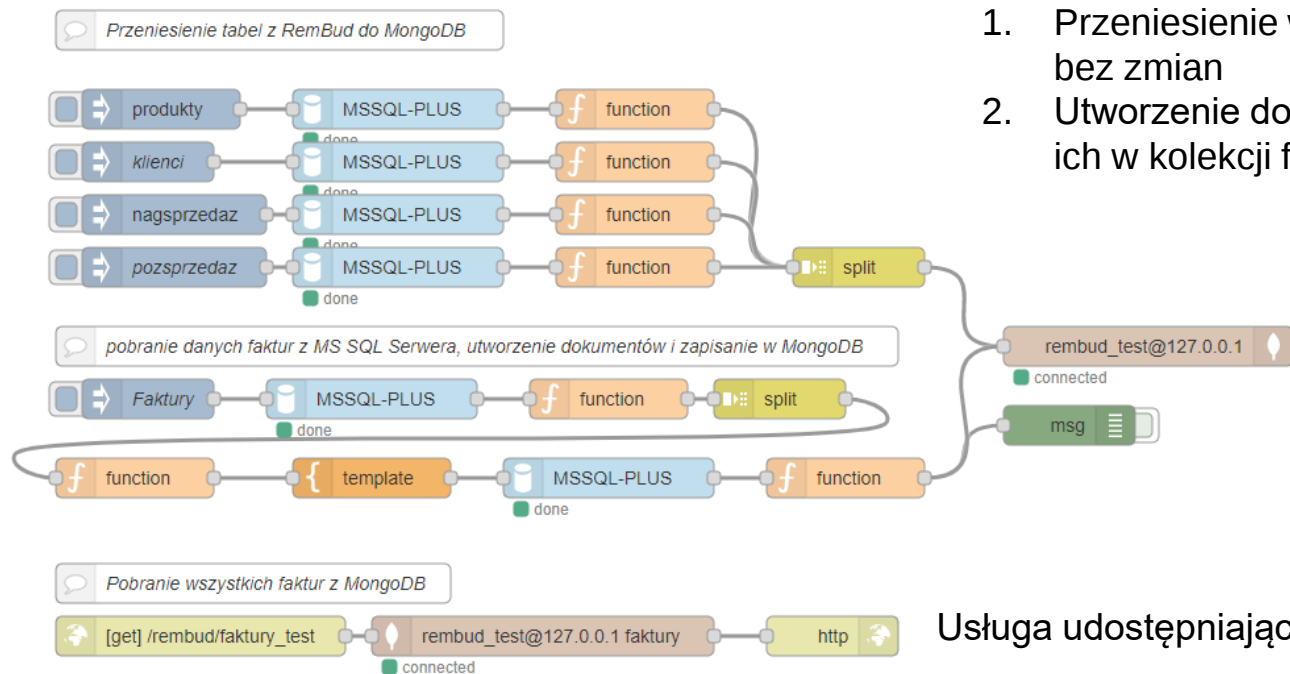
```
1 msg.collection = "faktury";
2 return msg;
```

Name: Zapis bez tablic

Setup Function Close

```
1 msg.collection = "faktury2";
2 msg.payload = {
3   "Nagłówek" : msg.payload.Nagłówek[0],
4   "Pozycje faktury" : msg.payload["Pozycje faktury"],
5   "Suma faktury" : msg.payload["Suma faktury"])[0]["suma faktury"]
6 }
7
8 return msg;
```

Przeniesienie danych z RemBud z serweraSQL do MongoDB



1. Przeniesienie wszystkich tabel do MongoDB bez zmian
2. Utworzenie dokumentów faktur i zapisanie ich w kolekcji faktury

Usługa udostępniająca kolekcję wszystkich faktur

Przykład wyszukiwania danych

The screenshot shows the MongoDB Compass interface. On the left, the sidebar displays the database structure: 8 DBS, 17 COLLECTIONS. The 'rembud3' database is selected, and the 'fakture' collection is highlighted. The main panel shows the 'rembud3.fakture' collection with 175 documents, a total size of 93.4KB, and 1 index. The 'Documents' tab is active, displaying a list of documents. The search query is: `{ "Pozycje_fakture.nr_fakture" : { $lt : 5 } }`. The results show three documents with their IDs and 'Suma_fakture' values: 154.23, 485.85, and 197.18.

MongoDB Compass - 192.168.0.211:27017/rembud3.fakture

Connect View Collection Help

Local

8 DBS 17 COLLECTIONS

☆ FAVORITE

HOST
192.168.0.211:27017

CLUSTER
Standalone

EDITION
MongoDB 4.4.1 Community

Filter your data

- > Pomiary
- > admin
- > config
- > local
- > rembud
- > rembud2
- > rembud3
 - fakture
 - klenci
 - produkty
 - test

rembud3.fakture Documents

DOCUMENTS 175 TOTAL SIZE 93.4KB AVG. SIZE 546B INDEXES 1 TOTAL SIZE 36.0KB AVG. SIZE 36.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER `{ "Pozycje_fakture.nr_fakture" : { $lt : 5 } }` **OPTIONS** **FIND** **RESET** **...**

PROJECT `{ "Suma_fakture" : { $sum : "$Suma_fakture.suma_fakture" } }`

SORT `{ field: -1 }` **MAX TIME MS** 60000

COLLATION `{ locale: 'simple' }` **SKIP** 0 **LIMIT** 0

VIEW **Displaying documents 1 - 3 of 3** **REFRESH**

`_id: ObjectId("608fc8fc24a99c04e23a9db8")`
`Suma_fakture: 154.23`

`_id: ObjectId("608fc8ff24a99c04e23a9dde")`
`Suma_fakture: 485.85`

`_id: ObjectId("608fc8ff24a99c04e23a9de7")`
`Suma_fakture: 197.18`

> _MongoSH Beta

Agregacja

```
db.collection.aggregate( [ { <stage> }, ... ] )
```

Tablica etapów (operacji) tworzących potok agregacji

Potok agregacji – Pipeline



Dane wyjściowe operacji są danymi wejściowymi kolejnej operacji

Wybrane operacje potoku agregacji

- **\$project** – określa jakie dane mają być umieszczone w dokumentach wyjściowych
- **\$much** – określa, które dokumenty są umieszczane w kolekcji wyjściowej - selekcja
- **\$limit** – ogranicza liczbę dokumentów przekazywanych do następnej operacji
- **\$skip** – pomija określoną liczbę dokumentów
- **\$unwind** – rozwija tablicę, generując jeden dokument dla każdego elementu tablicy
- **\$lookup** – dokonuje lewostronnego złączenia zewnętrznego kolekcji wejściowej z inną kolekcją
- **\$unionWith** – sumuje kolekcje
- **\$group** – grupuje dokument według podanego klucza
- **\$sort** – sortuje dokumenty w kolekcji
- **\$out** – zapisuje wynik działania potoku w kolekcji
- **\$merge** – dopisuje wynik działania potoku do kolekcji

Porównanie elementów zapytania SQL i operatorów agregacji

Klauzula SQL	Operator agregacji
SELECT	\$project
FROM	\$group, \$sum, \$min, \$max, \$avg, ...
JOIN	\$lookup, \$unwind
WHERE	\$much
GROUP BY	\$group
HAVING	\$much
ORDER BY	\$sort

```
db.collection.aggregate( [ {$match: ...}, {$group : ...}, {$lookup : ...}, {$match: ...}, ... , {$sort : ...} ] )
```

Funkcje agregujące (akumulujące) dane

Funkcja	Użycie
\$sum	{ \$sum: <expression> }, { \$sum: [<expression1>, <expression2> ...] } { "\$sum" : "\$pozycje.brutto" }, { "\$sum" : [3, 3, 4] }
\$max	
\$min	
\$avg	
\$first	
\$stdDevPop	

Wszystkie funkcje w tym matematyczne

$+$, $-$, $\times$, $/$

<https://docs.mongodb.com/manual/reference/operator/aggregation/>

Przykład agregacji – projekcja

The screenshot shows the MongoDB Compass interface for a database named 'rembud_test.faktery'. The left sidebar displays the database structure, including collections like 'Pomiary', 'admin', 'config', 'local', 'rembud', 'rembud2', 'rembud3', 'rembud_start', and 'rembud_test'. The 'rembud_test' collection is selected, showing a 'Projekcja2' aggregation pipeline.

The main panel shows the 'Aggregations' tab for the 'rembud_test.faktery' collection. The pipeline is currently empty, and the 'UPDATE VIEW' button is visible. The 'MODIFYING PIPELINE BACKING "REMBUD_TESTPROJEKCAJ2"' message is displayed.

Below the pipeline editor, the 'Preview of Documents in the Collection' section shows 98 documents. The 'Preview after \$project stage' section displays a sample of 20 documents, showing the result of the aggregation pipeline. The documents are structured as follows:

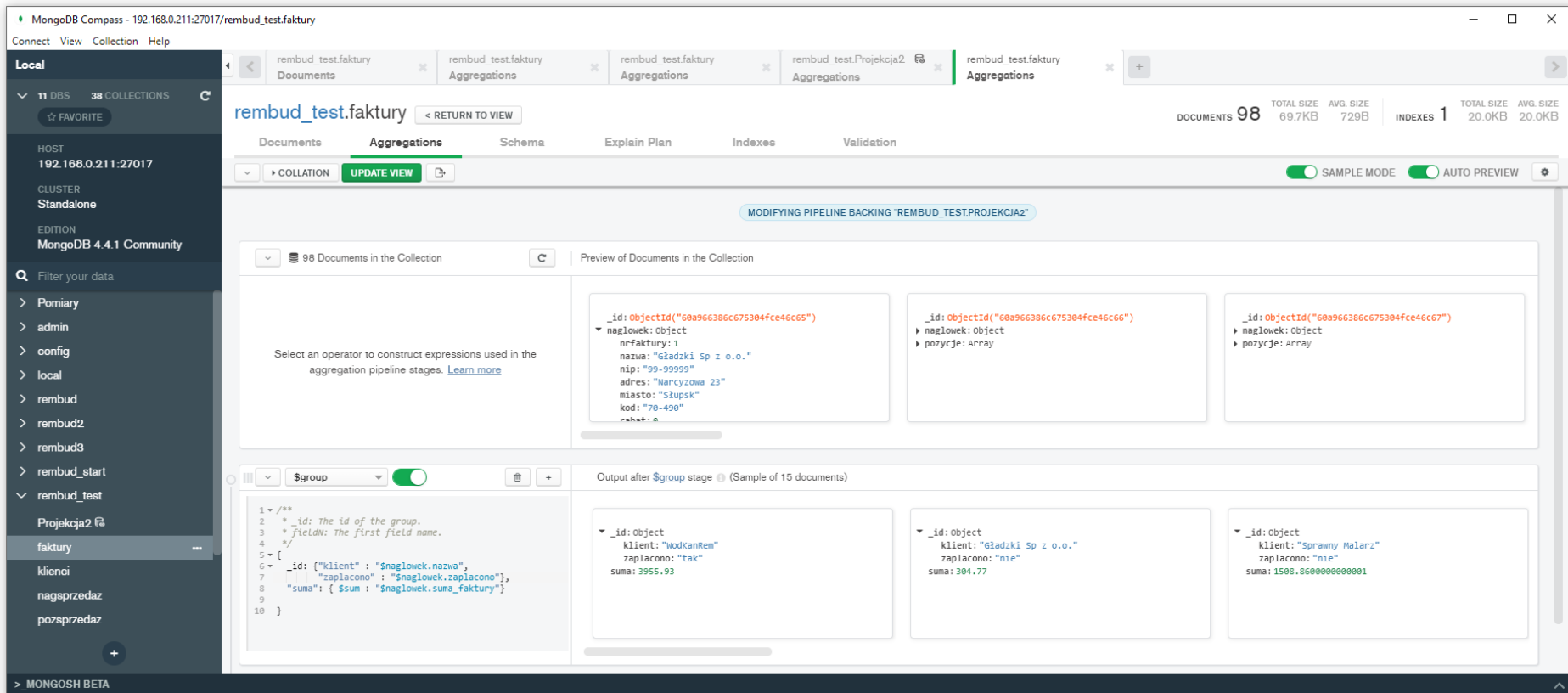
```
1 {
2   'nr': '5naglowek.nrfaktery',
3   'klient': '5naglowek.nazwa',
4   'suma z dokumentu': '5naglowek.suma_faktery',
5   'suma obliczona': {
6     $sum: '$spozycje.brutto'
7   }
8 }
```

The output shows two documents:

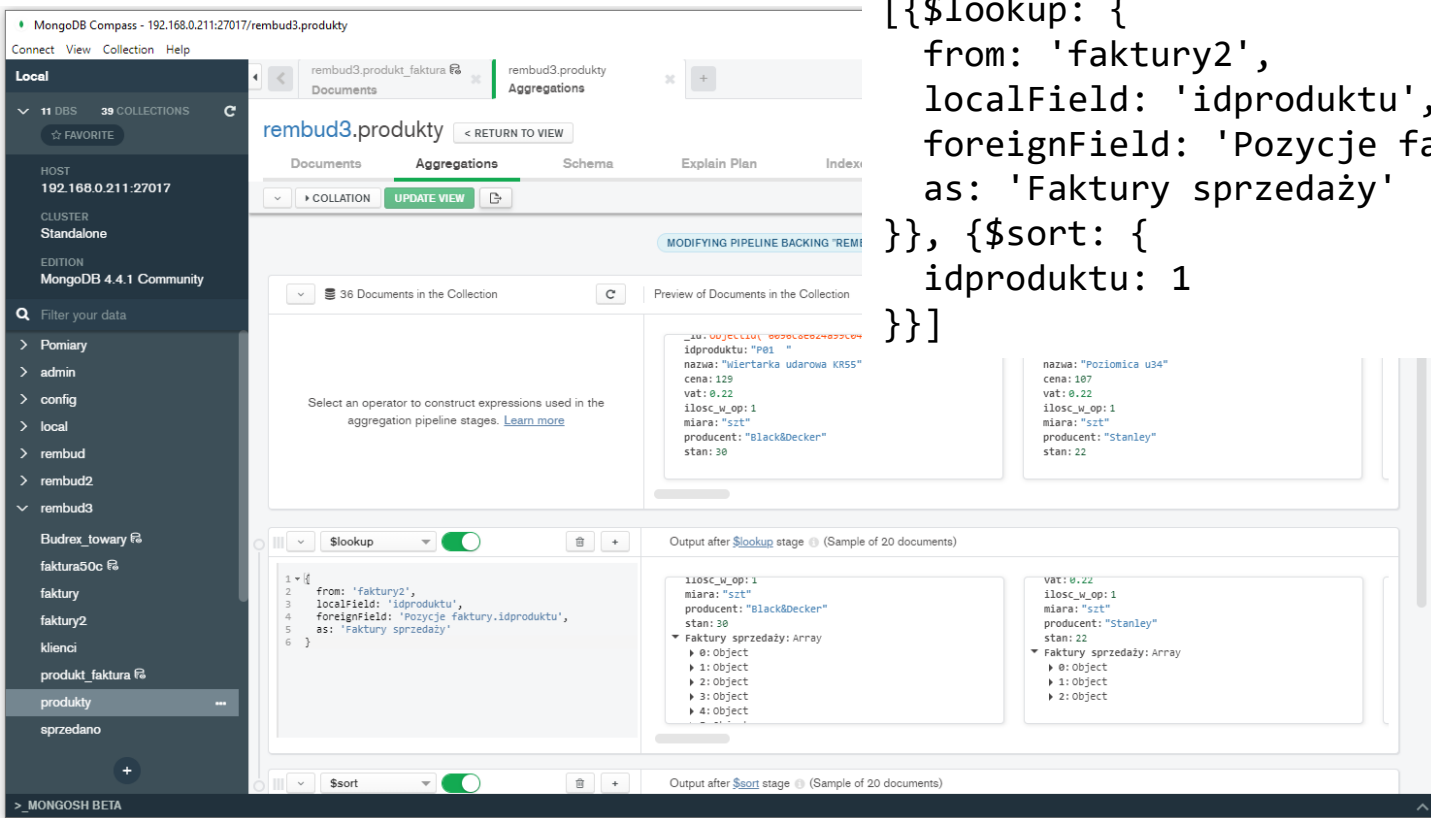
```
{
  "_id": "60a966386c675304fce46c65",
  "nr": 1,
  "klient": "Gładzki Sp z o.o.",
  "suma z dokumentu": 1398.27,
  "suma obliczona": 1398.27
}
```

```
{
  "_id": "60a966386c675304fce46c66",
  "nr": 2,
  "klient": "Posadzki Remonty",
  "suma z dokumentu": 154.23,
  "suma obliczona": 154.23
}
```

Przykład agregacji – grupowanie



Przykład agregacji – złączenie



The screenshot shows the MongoDB Compass interface. The left sidebar lists databases and collections. The main panel displays the 'rembud3.produkt' collection with an aggregation pipeline. The pipeline includes a \$lookup stage to join 'rembud3.faktury' into the 'Faktury sprzedaży' array. The output after the \$lookup stage shows documents with an array of sales facts. The output after the \$sort stage shows the documents sorted by the 'stan' field.

```
1 {
2   $lookup: {
3     from: 'faktury2',
4     localField: 'idproduktu',
5     foreignField: 'Pozycje faktury.idproduktu',
6     as: 'Faktury sprzedaży'
7   }
8 }
```

Output after \$lookup stage (Sample of 20 documents)

```
{
  "_id": "00000000000000000000000000000000",
  "idproduktu": "P01",
  "nazwa": "Wiertarka udarowa KR55",
  "cena": 129,
  "vat": 0.22,
  "ilosc_w_op": 1,
  "miara": "szt",
  "producent": "Black&Decker",
  "stan": 30,
  "Faktury sprzedaży": [
    {
      "idfaktury": "F01",
      "nazwa": "Poziomica u34",
      "cena": 107,
      "vat": 0.22,
      "ilosc_w_op": 1,
      "miara": "szt",
      "producent": "Stanley",
      "stan": 22
    }
  ]
}
```

Output after \$sort stage (Sample of 20 documents)

```
{
  "_id": "00000000000000000000000000000000",
  "idproduktu": "P01",
  "nazwa": "Wiertarka udarowa KR55",
  "cena": 129,
  "vat": 0.22,
  "ilosc_w_op": 1,
  "miara": "szt",
  "producent": "Black&Decker",
  "stan": 30,
  "Faktury sprzedaży": [
    {
      "idfaktury": "F01",
      "nazwa": "Poziomica u34",
      "cena": 107,
      "vat": 0.22,
      "ilosc_w_op": 1,
      "miara": "szt",
      "producent": "Stanley",
      "stan": 22
    }
  ]
}
```

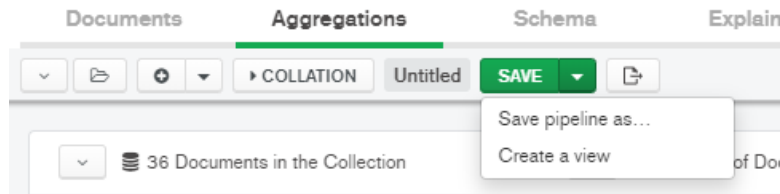
```
[{$lookup: {
  from: 'faktury2',
  localField: 'idproduktu',
  foreignField: 'Pozycje faktury.idproduktu',
  as: 'Faktury sprzedaży'
}}, {$sort: {
  idproduktu: 1
}}]
```

Tworzenie widoku

```
db.createView(  
  "<viewName>",  
  "<source>",  
  [<pipeline>],  
  { "collation" : { <collation> } } )
```

```
db.createView(  
  "<Nazwa widoku>",  
  "<kolekcja źródłowa>",  
  [<pipeline>],  
  { opcje : ... } )
```

Potok agregacji stworzony w MongoDB Compass można zapisać jako widok. Wynik działania potoku agregacji można dodać do istniejącej kolekcji lub zapisać w nowej kolekcji.



Walidacja dokumentów

MongoDB Compass - 192.168.0.211:27017/rembud_test.fakтуры

Connect View Collection Help

Local

11 DBS 38 COLLECTIONS

☆ FAVORITE

HOST
192.168.0.211:27017

CLUSTER
Standalone

EDITION
MongoDB 4.4.1 Community

Filter your data

Pomiary

admin

config

local

rembud

rembud2

rembud3

rembud_start

rembud_test

fakтуры

klienci

nagprzedaz

pozprzedaz

produkty

projekcja

rembud_wyklad

rembud_test.fakтуры

DOCUMENTS 98 TOTAL SIZE 69.7KB AVG. SIZE 729B INDEXES 1 TOTAL SIZE 20.0KB AVG. SIZE 20.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

Validation Action ERROR Validation Level STRICT

```
1 {
2   "naglowek.nr_fakтуры" : { $type : "int" },
3   "naglowek.nr_fakтуры" : { $type : "int" },
4
5   "naglowek.miasto" : { $type : "string" },
6
7 }
```

Validation modified

CANCEL UPDATE

✓ Sample Document That Passed Validation

✗ Sample Document That Failed Validation

Sample Document That Passed Validation

```
{
  "_id": ObjectId("60a966386c675304fce46c65"),
  "naglowek": Object,
  "nr_fakтуры": 1,
  "nazwa": "Gładzki Sp z o.o.",
  "nip": "99-99999",
  "adres": "Warcyzowa 23",
  "miasto": "Ślupsk",
  "kod": "78-498",
  "rachunek": 0
}
```

Sample Document That Failed Validation

```
{
  "_id": ObjectId("60a966386c675304fce46c6a"),
  "naglowek": Object,
  "nr_fakтуры": 8,
  "nazwa": "Posadzki Remonty",
  "nip": "99-88732",
  "adres": "Różana 13",
  "miasto": "Gdańsk",
  "kod": "78-231",
  "rachunek": 0
}
```

Dla kolekcji można określić warunki jakie muszą spełniać dokumenty dodawane do kolekcji

<https://docs.mongodb.com/manual/core/schema-validation/>.