Stacja paliw

Opis

Sieci stacji paliw potrzebna jest baza danych do przechowywania informacji o sprzedaży, magazynowaniu i logistyce paliw naftowych, pracownikach, produktach sprzedaży detalicznej oraz nieruchomościach.

Nieruchomości muszą mieć określony typ: stacja lub magazyn, adres oraz kierownika. Pracownicy muszą mieć przypisane miejsce zatrudnienia oraz stanowisko. Każdy pracownik ma służbowy e-mail, który służy również jako identyfikator przy połączeniu z zewnętrznymi systemami. Pracownicy magazynów mogą tworzyć i wysyłać transporty. Pracownicy stacji paliw mogą zamawiać paliwa, zarządzać sprzedażą. Zamówienia muszą mieć określony towar oraz ilość, magazyn oraz stację dostawy oraz pracowników zamawiających i realizujących zamówienie. Nieruchomości muszą mieć określone stany magazynowe, zawierające towar oraz jego ilość. Na danych stacjach pracownicy muszą mieć możliwość określenia cen surowców.

Założenia bazy

Pracownicy

- muszą mieć określone imię i nazwisko
- muszą mieć określone stanowisko
- muszą mieć określony email służbowy
- muszą mieć określoną stację lub magazyn zatrudnienia

Pracownicy magazynów

- muszą mieć możliwość podglądu zamówień do realizacji
- muszą mieć możliwość zmiany statusu zamówienia na "w realizacji"

Kierowcy

muszą mieć możliwość zmiany statusu zamówienia na "w dostawie"

Pracownicy stacji

- muszą mieć możliwość składania zamówień na konkretne towary
- muszą mieć możliwość zmiany statusu zamówienia na "zakończone"

Kierownicy stacji

muszą mieć możliwość zmieniania cen towarów

Stacje

muszą mieć określony typ: stacja lub magazyn

- muszą mieć określony adres
- muszą mieć określonego kierownika
- muszą mieć określone stany magazynowe konkretnych towarów
- muszą mieć określone ceny konkretnych towarów

Magazyny

- muszą mieć określony typ: stacja lub magazyn
- muszą mieć określony adres
- muszą mieć określonego kierownika
- muszą mieć określone stany magazynowe konkretnych towarów

Towary

- muszą mieć określone typy paliwa
- muszą mieć określone nazwy marketingowe

Zamówienia

- muszą mieć określony status
- muszą mieć określoną listę towarów
- muszą mieć określony magazyn realizujący zamówienie
- muszą mieć określoną stację docelową
- muszą mieć określonego pracownika realizującego zamówienie
- muszą mieć określonego pracownika składającego zamówienie
- muszą mieć określonego kierowcę przewożącego zamówienie

Kolekcje bazy danych

gas_stationIDproducts[]					
r					
de					
y					
)					

Nazwa kolekcji	warehouses				
Pola					
warehousesID					
• products[]					
o type					
o name					
o stock					
 address 					
o region	1				
o city					
o street					
o numbe	er				
o zip_co	ode				
o countr	ry				

Nazwa kolekcji	workers					
Pola						
 workesrsID 						
• name						
• surname						
• email						
place_of_employment						
o type						
o id						
job_position	• job_position					

Nazwa kolekcji			orders
Pola			
•	ordersII		
•	product		
	0	type	
	0	name	
		stock	
•	origin		-
	0	storages	
	0	address	
		•	region
		•	city
		•	street
		•	number
		•	zip_code
		•	country
•	target		' ID
	0	gas_stat address	ionsiD
		•	region
		•	city
		•	street
		•	number
		•	zip_code
		•	country
•	order_o	wner	
	0	workers	ID
	0	names	
	0	surname	
	0	email	

Nazwa kolekcji		orders_details					
Pola	Pola						
• order_ic • order_st • order_st • drivers[• o • storekee	atus] workesi name surname email	rsID e sID					

Zapytania do bazy danych

Stacje z niskim stanem magazynowym któregoś paliwa

db.gas_stations.find({ "products": { \$elemMatch:{ stock:{\$lte:3000}}}})

Stacje na ulicy Nadbystrzyckiej w Lublinie

db.gas stations.find({"address.city":'Lublin', "address.street":'Nadbystrzycka'})

pracownicy o nazwisku Kowalski

db.workers.find({surname:'Kowalski'})

```
> db.workers.find({surname:'Kowalski'})

< { _id: ObjectId("6248a4105307713668a0fa7f"),
    name: 'Jan',
    surname: 'Kowalski',
    email: 'jk@gmail.com',
    place_of_employment: { type: 'gas_station', id: '624888f1f4fb08e41ea6e3a3' },
    job_position: 'gas_station_worker' }</pre>
```

pracownik o najniższym ID

 $db.workers.find().sort(\{``_id":1\}).limit(1)$

zamówienia w transporcie

db.orders details.find({"order status":'in progress'})

zamówienia pracownika Jan kowalski

db.orders.find({"order_owner.name":'Jan',"order_owner.surname":'Kowalski'})

```
> db.orders.find({"order_owner.name":'Jan',"order_owner.surname":'Kowalski'})
< { _id: ObjectId("6248a8f35307713668a0fa8c"),
        type: 'Diesel',
        stock: Decimal128("5000") },
      { name: 'Diesel+', type: 'Diesel', stock: Decimal128("4000") } ],
    { id: '6248896ef4fb08e41ea6e3a8',
       { country: 'Poland',
        region: 'Mazowieckie',
         city: 'Warsaw',
         street: 'Marszałkowska',
         zip code: '00-123' } },
    { id: '624888f1f4fb08e41ea6e3a4',
       { country: 'Poland',
        region: 'Lubelskie',
         city: 'Lublin',
         street: 'Nadbystrzycka',
        number: '24',
         zip code: '01-123' } },
    { id: '6248a4105307713668a0fa7f',
      name: 'Jan',
      surname: 'Kowalski',
      email: 'jk@gmail.com' } }
 { id: ObjectId("6248a8f35307713668a0fa8d"),
    [ { name: 'Super Diesel',
        type: 'Diesel',
        stock: Decimal128("3000") },
```

Stacje w województwie Mazowieckim

 $db.gas_stations.find(\{``address.region": `Mazowieckie', ``address.country": "Poland"\})$

```
> db.gas stations.find({"address.region":"Mazowieckie", "address.country":"Poland"})
< { id: ObjectId("624888f1f4fb08e41ea6e3a3"),</pre>
     [ { name: 'Super Diesel',
         type: 'Diesel',
         stock: Decimal128("5000"),
         price: Decimal128("6.70000000000000") },
       { name: 'Diesel+',
         type: 'Diesel',
        stock: Decimal128("4000"),
        price: Decimal128("6.70") } ],
     { country: 'Poland',
       region: 'Mazowieckie',
      city: 'Warsaw',
      street: 'Marszałkowska',
      number: '51',
       zip code: '00-123' } }
stacja paliw > db.gas stations.find({"address.region": "Mazowieckie", "address.country": "Poland"})
```

suma paliw w zamówieniach

```
 db.orders.aggregate([ \{ \$group: \{ \_id: "\$\_id", total: \{ "\$\$sum": \{ "\$\$sum": "\$products.stock" \} \} \} \}, \{ \$\$sort: \{ total: -1 \} \}])
```

łączna ilość paliw na stacjach

liczba pracowników o danych nazwiskach

db.workers.aggregate([{\$group: {_id: "\$surname", liczba: {\$sum: 1}}}])

```
> db.workers.aggregate([{$group : {_id: "$surname", liczba: {$sum : 1}}}])
< { _id: 'Kowalski', liczba: 1 }
    {_id: 'Nowak', liczba: 1 }
    {_id: 'Małysz', liczba: 1 }</pre>
```

Uaktualnienia danych

Zmiana nazwiska pracownika

```
db.workers.update(
{ "name": "Michał", "surname": "Małysz"},
{$set: {"surname": "Domański"}}
)
```

Przed:

```
_id: ObjectId('6248a4105307713668a0fa81')
name: "Michal"
surname: "Malysz"
email: "mm@gmail.com"

> place_of_employment: Object
    type: "warehouse"
    id: "6248896ef4fb08e41ea6e3a8"
    job_position: "storekeeper"
```

```
_id: ObjectId('6248a4105307713668a0fa81')
name: "Michał"
surname: "Domański"
email: "mm@gmail.com"

v place_of_employment: Object
    type: "warehouse"
    id: "6248896ef4fb08e41ea6e3a8"
    job_position: "storekeeper"
```

Zmiana statusu zamówienia na closed

```
db.orders_details.update(
{ "order_id": "6248a8f35307713668a0fa8c"},
{$set: {"order_status": "closed"}}
)
```

Przed:

```
_id: ObjectId('6248abe75307713668a0fa9b')
    order_id: "6248a8f35307713668a0fa8c"
    order_status: "in_progress"
    v drivers: Array
    > 0: Object
    v storekeeper: Object
    id: "6248a4105307713668a0fa81"
    name: "Michał"
    surname: "Małysz"
    email: "mm@gmail.com"
```

Zmiana maila pracownika

```
db.workers.update(
{ "name": "Adam", "surname": "Nowak"},
{$set: {"email": "adamNowak@gmail.com"}}
)
```

Przed:

```
_id: ObjectId('6248a4105307713668a0fa80')
name: "Adam"
surname: "Nowak"
email: "an@gmail.com"

v place_of_employment: Object
    type: "warehouse"
    id: "6248896ef4fb08e41ea6e3a8"
    job_position: "driver"
```

```
_id: ObjectId('6248a4105307713668a0fa80')
name: "Adam"
surname: "Nowak"
email: "adamNowak@gmail.com"

v place_of_employment: Object
    type: "warehouse"
    id: "6248896ef4fb08e41ea6e3a8"
    job_position: "driver"
```

Zmiana magazyniera w szczegółach zamówienia

```
db.orders_details.update(
{ "order_id": "6248a8f35307713668a0fa8c"},
{$set: { "storekeeper": {"id": "6248a4105307713668a0fa7f", "name": "Jan", "surname": "Kowalski",
"email":"jk@gmail.com"} }}
)
```

Przed:

```
_id: ObjectId('6248abe75307713668a0fa9b')
order_id: "6248a8f35307713668a0fa8c"
order_status: "closed"

v drivers: Array

v 0: Object
    id: "6248a4105307713668a0fa80"
    name: "Adam"
    surname: "Nowak"
    email: "an@gmail.com"

v storekeeper: Object
    id: "6248a4105307713668a0fa81"
    name: "Michał"
    surname: "Małysz"
    email: "mm@gmail.com"
```

```
_id: ObjectId('6248abe75307713668a0fa9b')
order_id: "6248a8f35307713668a0fa8c"
order_status: "closed"

v drivers: Array

v 0: Object
    id: "6248a4105307713668a0fa80"
    name: "Adam"
    surname: "Nowak"
    email: "an@gmail.com"

v storekeeper: Object
    id: "6248a4105307713668a0fa7f"
    name: "Jan"
    surname: "Kowalski"
    email: "jk@gmail.com"
```

Zmiana stanu magazynu

```
db.warehouses.update(
{"address.city":"Lublin"},
{$set: {"products.0.stock":0}}
)
```

Przed:

```
_id: ObjectId('6248896ef4fb08e41ea6e3a9')

> products: Array

> 0: Object

    name: "Super 95"

    type: "95"

    stock: 4000

> 1: Object

> address: Object

    country: "Poland"

    region: "Lubelskie"

    city: "Lublin"

    street: "Nadbystrzycka"

    number: "24"

    zip_code: "01-123"
```

```
_id: ObjectId('6248896ef4fb08e41ea6e3a9')

> products: Array

> 0: Object

    name: "Super 95"

    type: "95"

    stock: 0

> 1: Object

> address: Object

    country: "Poland"

    region: "Lubelskie"

    city: "Lublin"

    street: "Nadbystrzycka"

    number: "24"

    zip_code: "01-123"
```

Użytkownicy

Stworzenie admina

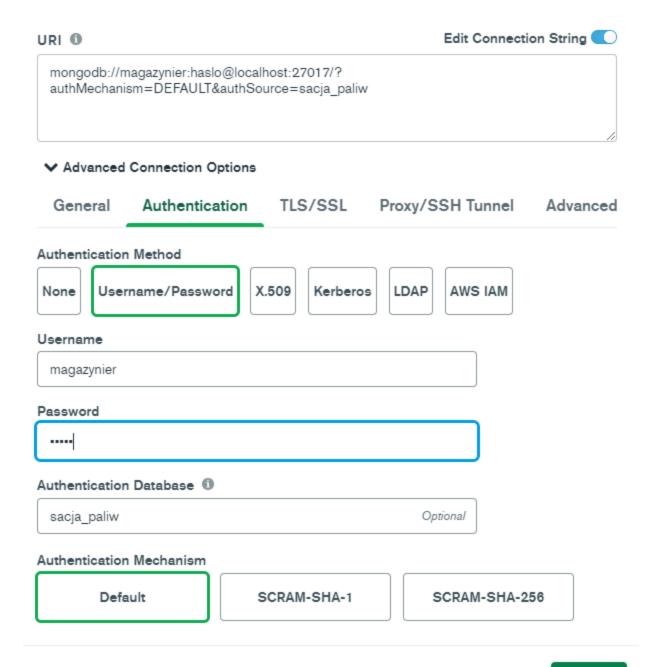
```
db.createUser(
    {
        user: "admin",
        pwd: passwordPrompt(),
        roles: [
            { role: "userAdminAnyDatabase", db: "admin" },
            { role: "readWriteAnyDatabase", db: "admin" }
        }
    }
}
```

Stworzenie magazyniera

```
db.createUser(
    {
      user: "magazynier",
      pwd: "haslo",
      roles: [ { role: "readWrite", db: "stacja_paliw" } ]
    }
}

> db.createUser(
    {
      user: "magazynier",
      pwd: "haslo",
      roles: [ { role: "readWrite", db: "stacja_paliw" } ]
    }
    }
}

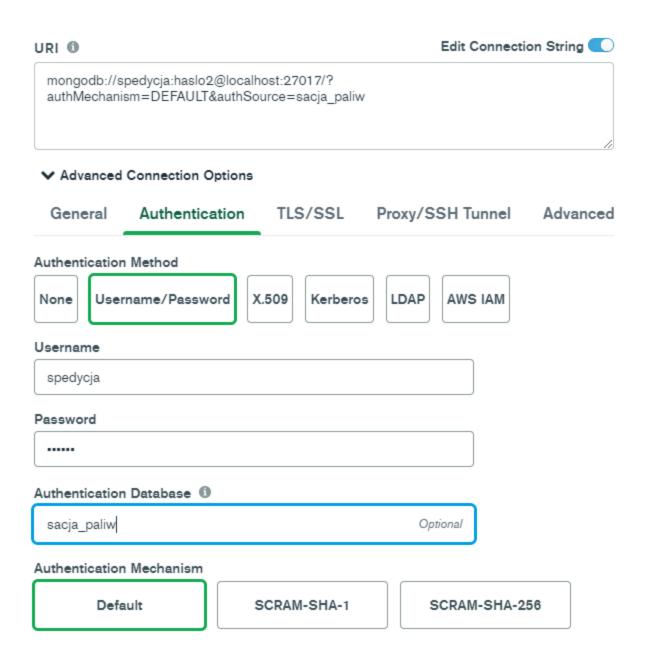
( { ok: 1 }
```



Connect

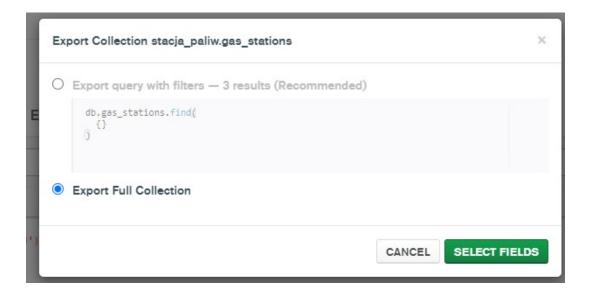
Stworzenie konta dla spedycji

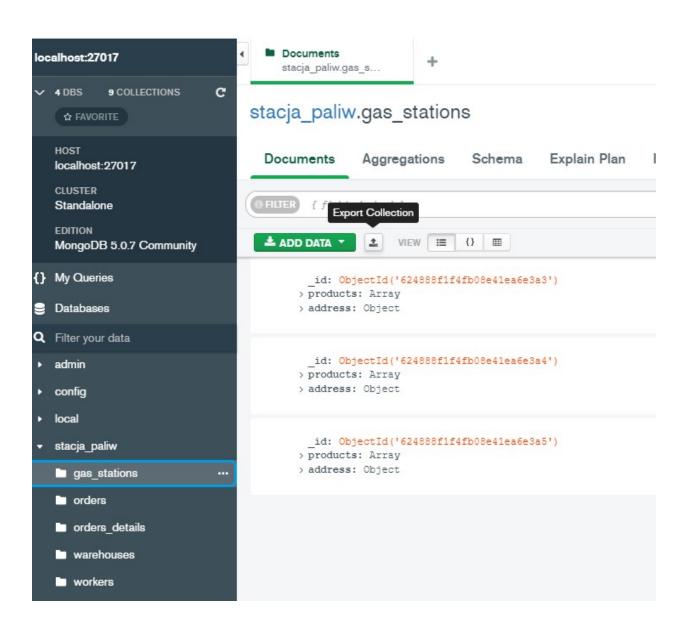
```
db.createUser(
    {
      user: "spedycja",
      pwd: "haslo2",
      roles: [ { role: "read", db: "stacja_paliw" } ]
    }
)
```



Eksport/import kopii bazy danych

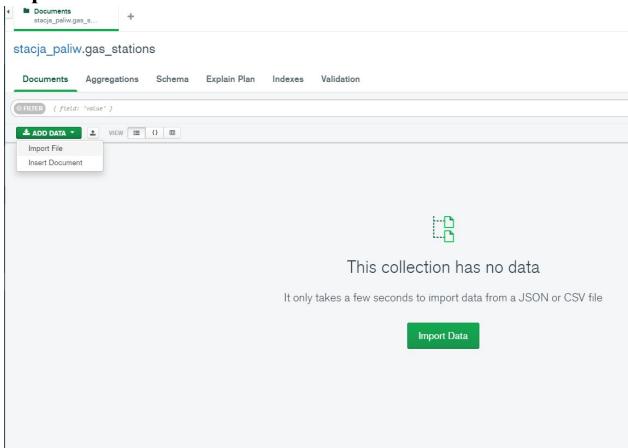
Export:

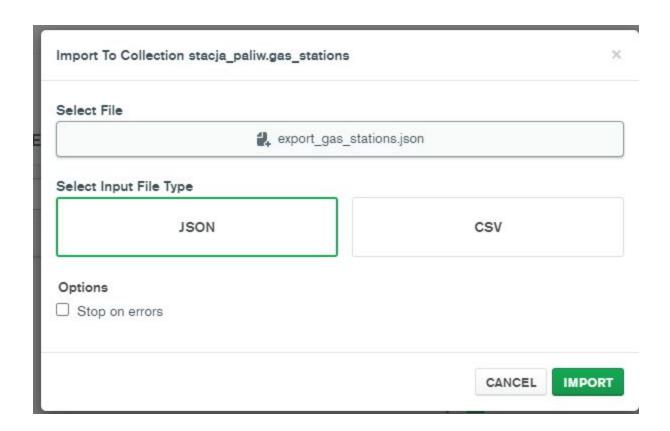






Import:





stacja_paliw.gas_stations Aggregations Schema Explain Plan Indexes Documents OFILIER { field: 'value' } ▲ ADD DATA ▼ VIEW : {} _id: ObjectId('624888f1f4fb08e41ea6e3a3') > products: Array > address: Object id: ObjectId('624888f1f4fb08e41ea6e3a4') > products: Array > address: Object _id: ObjectId('624888f1f4fb08e41ea6e3a5') > products: Array > address: Object

Zapytania w LiteDB

Wyszukanie dwóch pierwszych pracowników posortowanych alfabetycznie

SELECT \$ FROM workers ORDER BY surname LIMIT 2

```
SELECT $ FROM workers ORDER BY surname LIMIT 2
      Text
Grid
             Parameters
  /* 1 */
    "_id": {"$oid": "6248a4105307713668a0fa7f"},
    "name": "Jan",
    "surname": "Kowalski",
    "email": "jk@gmail.com",
    "place_of_employment":
      "type": "gas_station",
      "id": "624888f1f4fb08e41ea6e3a3"
    "job_position": "gas_station_worker"
    "_id": {"$oid": "6248a4105307713668a0fa81"},
    "name": "Michał",
    "surname": "Małysz",
    "email": "mm@gmail.com",
    "place_of_employment":
      "type": "warehouse",
      "id": "6248896ef4fb08e41ea6e3a8"
    "job_position": "storekeeper"
```

Wyszukanie produktów w magazynach w mieście Lublin

SELECT products FROM warehouses WHERE address.city='Lublin'

Wyszukanie zamówień z Polski do Polski

SELECT \$ FROM orders WHERE target.address.country='Poland' AND origin.address.country='Poland'



Wyszukanie id stacji gdzie jako pierwszy produkt jest Diesel

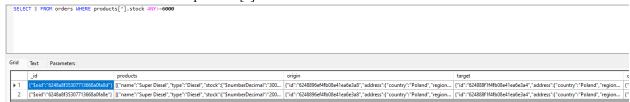
SELECT id FROM gas stations WHERE products[0].type='Diesel'

```
Grid Text Parameters

/* 1 */
{
    "_id": {"$oid": "624888f1f4fb08e41ea6e3a3"}
}
/* 2 */
{
    "_id": {"$oid": "624888f1f4fb08e41ea6e3a5"}
}
```

Wyszukanie zamówień, gdzie występuje produkt w ilości większej lub równej 6000

SELECT \$ FROM orders WHERE products[*].stock ANY>=6000



Zliczenie ilości pracowników z podziałem na miejsce pracy

SELECT @key AS place_of_employment_type, COUNT(*) FROM workers GROUP BY place_of_employment.type

```
SELECT @key AS place_of_employment_type, COUNT(*) FROM workers GROUP BY place_of_employment.type

Grid    Text    Parameters

/* 1 */
{
        "place_of_employment_type": "gas_station",
        "expr": 1
}
/* 2 */
{
        "place_of_employment_type": "warehouse",
        "expr": 2
}
```

Zliczenie stacji benzynowych w danym kraju oraz wyświetlanie produktów w tych krajach

SELECT @key AS country, COUNT(*) AS total, *.products AS products FROM gas_stations GROUP BY address.country

```
Grid
               Parameters
    "country": "Poland",
    "total": 3,
     "products":
      Γ
          "name": "Super Diesel",
           "type": "Diesel",
          "stock": {"$numberDecimal": "5000"},
          "price": {"$numberDecimal": "6.70"}
          "name": "Diesel+",
          "type": "Diesel",
          "stock": {"$numberDecimal": "4000"},
          "price": {"$numberDecimal": "6.70"}
      ],
      Γ
          "name": "Super 95",
           "type": "95",
          "stock": {"$numberDecimal": "4000"},
          "price": {"$numberDecimal": "5.70"}
          "name": "Diesel+",
          "type": "Diesel",
          "stock": {"$numberDecimal": "6000"},
          "price": {"$numberDecimal": "6.62"}
      ],
      "name": "Super Diesel",
          "type": "Diesel",
          "stock": {"$numberDecimal": "3000"},
          "price": {"$numberDecimal": "6.20"}
          "name": "Diesel+",
          "type": "Diesel",
          "stock": {"$numberDecimal": "5600"},
          "price": {"$numberDecimal": "7.70"}
          "name": "Super 95",
          "type": "95",
          "stock": {"$numberDecimal": "3000"},
           "price": {"$numberDecimal": "6.70"}
    ]
```

Wyświetlanie średniej ceny produktów w Polsce z podziałem na miasta

SELECT @key AS city, AVG(*.products[*].price) AS average_of_products_price FROM gas_stations WHERE address.country="Poland" GROUP BY address.city

Wyświetlenie liczby paliw z każdej stacji jeśli liczba paliw będzie większa lub równa 2

SELECT @key as id, COUNT(*.products[*]) AS products FROM gas_stations GROUP BY _id HAVING COUNT(*.products[*])>=2

```
SELECT @key as id, COUNT(*.products[*]) AS products FROM gas_stations GROUP BY _id HAVING COUNT(*.products[*])>=2

Grid Text Parameters

/* 1 */
{
    "id": {"$oid": "624888f1f4fb08e41ea6e3a4"},
    "products": 2
}
/* 2 */
{
    "id": {"$oid": "624888f1f4fb08e41ea6e3a5"},
    "products": 2
}
/* 3 */
{
    "id": {"$oid": "624888f1f4fb08e41ea6e3a5"},
    "products": 3
}
```

Zliczenie ilości produktów z podziałem na region

SELECT @key AS region, SUM(*.products[*].stock) FROM warehouses GROUP BY address.region

```
SELECT @key AS region, SUM(*.products[*].stock) FROM warehouses GROUP BY address.region
```

Grid Text Parameters

```
/* 1 */
{
    "region": "Lubelskie",
    "products": {"$numberDecimal": "10000"}
}
/* 2 */
{
    "region": "Mazowieckie",
    "products": {"$numberDecimal": "9000"}
}
/* 3 */
{
    "region": "Wielkopolskie",
    "products": {"$numberDecimal": "11600"}
}
```