

LABORATORIUM NIERELACYJNE BAZY DANYCH

**Data wykonania
ćwiczenia:**

06.05.2023

Rok studiów:

3

Semestr:

6

Grupa studencka:

2

Grupa laboratoryjna:

2B

Ćwiczenie nr.

7

Temat: Obliczenia rozproszone. MapReduce w MONGODB

Osoby wykonujące ćwiczenia:

1. Igor Gawłowicz

Katedra Informatyki i Automatyki

Obliczenia rozproszone. MapReduce w MONGODB

Utwórz kolekcję, dla której będą przechowywane dokumenty źródłowe MapReduce

```
> db.createCollection("selling")
{ ok: 1 }
```

Wypełnij kolekcję danymi.

```
test> db.selling.insert({name: 'Nexus One', date: new Date(2013, 0, 20, 12, 00)})
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany,
or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6638cb1e57c610e377160566') }
}
test> db.selling.insert({name: 'Nexus One', date: new Date(2013, 0, 20, 13, 00)})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6638cb1e57c610e377160567') }
}
test> db.selling.insert({name: 'iPhone 4', date: new Date(2013, 0, 20, 13, 30)})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6638cb1f57c610e377160568') }
}
test> db.selling.insert({name: 'Nexus One', date: new Date(2013, 0, 20, 14, 00)})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6638cb1f57c610e377160569') }
}
test> db.selling.insert({name: 'iPhone 4', date: new Date(2013, 0, 21, 15, 30)})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6638cb1f57c610e37716056a') }
}
test> db.selling.insert({name: 'iPhone 4', date: new Date(2013, 0, 21, 15, 40)})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6638cb1f57c610e37716056b') }
}
test> db.selling.insert({name: 'Nexus One', date: new Date(2013, 0, 21, 16, 20)})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6638cb1f57c610e37716056c') }
}
test> db.selling.insert({name: 'iPhone 4', date: new Date(2013, 0, 21, 17, 00)})
{
  acknowledged: true,
```

```

    insertedIds: { '0': ObjectId('6638cb2057c610e37716056d') }
  }
test> db.selling.insert({name: 'Nexus One', date: new Date(2013, 0, 21, 18, 00)})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6638cb2057c610e37716056e') }
}
test> db.selling.insert({name: 'Nexus One', date: new Date(2013, 0, 22, 19, 00)})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6638cb2457c610e37716056f') }
}

```

Utwórz funkcję Map w konsoli (konsola umożliwia wprowadzanie konstrukcji wielokreskowych).

```

test> var map = function() {
... var key = {
... name: this.name,
... year: this.date.getFullYear(),
... month: this.date.getMonth(),
... day: this.date.getDate()
... };
... emit(key, {count: 1}); }

```

Utwórz funkcję Reduce w konsoli

```

test> var reduce = function(key, values) {
... var sum = 0;
... values.forEach(function(value) {
... sum += value['count'];
... });
... return {count: sum};
... };

```

Wykonaj polecenie MapReduce na kolekcji „selling”. Określ parametr „inline: 1”, aby wyprowadzić wynik do konsoli.

```

test> db.selling.mapReduce(map, reduce, {out: {inline: 1}})
DeprecationWarning: Collection.mapReduce() is deprecated. Use an aggregation
instead.
See https://docs.mongodb.com/manual/core/map-reduce for details.
{
  results: [
    {
      _id: { name: 'Nexus One', year: 2013, month: 0, day: 20 },
      value: { count: 3 }
    },
  ],
}

```

```

{
  _id: { name: 'iPhone 4', year: 2013, month: 0, day: 21 },
  value: { count: 3 }
},
{
  _id: { name: 'Nexus One', year: 2013, month: 0, day: 21 },
  value: { count: 2 }
},
{
  _id: { name: 'Nexus One', year: 2013, month: 0, day: 22 },
  value: { count: 1 }
},
{
  _id: { name: 'iPhone 4', year: 2013, month: 0, day: 20 },
  value: { count: 1 }
}
],
ok: 1
}

```

Funkcja jest przestarzała według mongo ale wszystko działa tak jak powinno działać.

Odczytaj wynik działania mapReduce

```

test> db.selling.mapReduce(map, reduce, {out: {inline: 1}})
{
  results: [
    {
      _id: { name: 'Nexus One', year: 2013, month: 0, day: 22 },
      value: { count: 1 }
    },
    {
      _id: { name: 'iPhone 4', year: 2013, month: 0, day: 20 },
      value: { count: 1 }
    },
    {
      _id: { name: 'iPhone 4', year: 2013, month: 0, day: 21 },
      value: { count: 3 }
    },
    {
      _id: { name: 'Nexus One', year: 2013, month: 0, day: 20 },
      value: { count: 3 }
    },
    {
      _id: { name: 'Nexus One', year: 2013, month: 0, day: 21 },
      value: { count: 2 }
    }
  ],
  ok: 1
}

```

Wykonaj polecenie MapReduce z kolekcja „selling”. Aby wyprowadzić wynik do kolekcji, parametr „out” przyjmuje nazwę wynikowej kolekcji.

```
test> db.selling.mapReduce(map, reduce, {out: 'sellingResult'})
{ result: 'sellingResult', ok: 1 }
```

Sprawdź, czy funkcja działała bez błędów i czy wynik jest zgodny z oczekiwaniami.

Wszystkie wyniki zgadzają się z oczekiwaniami

Zapytaj o listę dokumentów z kolekcji „sellingResult”:

```
test> db.sellingResult.find()
[
  {
    _id: { name: 'iPhone 4', year: 2013, month: 0, day: 20 },
    value: { count: 1 }
  },
  {
    _id: { name: 'iPhone 4', year: 2013, month: 0, day: 21 },
    value: { count: 3 }
  },
  {
    _id: { name: 'Nexus One', year: 2013, month: 0, day: 21 },
    value: { count: 2 }
  },
  {
    _id: { name: 'Nexus One', year: 2013, month: 0, day: 22 },
    value: { count: 1 }
  },
  {
    _id: { name: 'Nexus One', year: 2013, month: 0, day: 20 },
    value: { count: 3 }
  }
]
```

Porównaj listę otrzymanych dokumentów z oczekiwanym wynikiem:

```
{ "_id" : { "name" : "Nexus One", "year" : 2013, "month" : 0, "day" : 20 },
  "value" : { "count" : 3 } }
{ "_id" : { "name" : "Nexus One", "year" : 2013, "month" : 0, "day" : 21 },
  "value" : { "count" : 2 } }
{ "_id" : { "name" : "Nexus One", "year" : 2013, "month" : 0, "day" : 22 },
  "value" : { "count" : 1 } }
{ "_id" : { "name" : "iPhone 4", "year" : 2013, "month" : 0, "day" : 20 }, "value"
: { "count" : 1 } }
{ "_id" : { "name" : "iPhone 4", "year" : 2013, "month" : 0, "day" : 21 }, "value"
: { "count" : 3 } }
```

Kolejność rekordów jest inna wygląda na losową, poza tym wynik bez zmian

1. Utwórz kolekcję dokumentów, aby przetworzyć je za pomocą MapReduce

Na potrzeby tego zadania utworzymy sobie kolekcję **dokumenty**

```
db.createCollection("dokumenty")
```

2. Wypełnij kolekcję dokumentami.

Następnie wprowadzimy 10 dokumentów do tej kolekcji

```
test> db.dokumenty.insertMany([
...   { _id: 1, name: "Document 1", type: "Type A", value: 100 },
...   { _id: 2, name: "Document 2", type: "Type B", value: 200 },
...   { _id: 3, name: "Document 3", type: "Type A", value: 150 },
...   { _id: 4, name: "Document 4", type: "Type C", value: 120 },
...   { _id: 5, name: "Document 5", type: "Type B", value: 180 },
...   { _id: 6, name: "Document 6", type: "Type A", value: 220 },
...   { _id: 7, name: "Document 7", type: "Type C", value: 130 },
...   { _id: 8, name: "Document 8", type: "Type B", value: 250 },
...   { _id: 9, name: "Document 9", type: "Type A", value: 170 },
...   { _id: 10, name: "Document 10", type: "Type C", value: 190 }
... ])
```

3. Wykonaj przetwarzanie kolekcję przy użyciu modelu przetwarzania rozproszonego MapReduce.

Utworzymy teraz nową funkcję mapującą

```
test> var mapFunction = function() {
...   emit(this.type, this.value);
... };
```

Teraz funkcję redukującą

```
test> var reduceFunction = function(key, values) {
...   return Array.sum(values);
... };
```

Wykonamy teraz operację

```
test> db.dokumenty.mapReduce(
...   mapFunction,
```

```
...   reduceFunction,  
...   { out: "wyniki" }  
... );  
{ result: 'wyniki', ok: 1 }
```