

Сидоренко Катерина, ФБ-11мп

Lab 7 - Написання MapReduce на прикладі MongoDB

Коллекции orders и items

```
mydb> db.orders.find()
[
  {
    _id: ObjectId("61ba5c214f2a97bda1f918ec"),
    order_number: 3000,
    date: ISODate("2021-04-13T00:00:00.000Z"),
    total_sum: 700.7,
    customer: {
      name: 'Andrii',
      surname: 'Rodionov',
      phones: [ 9876543, 1234567 ],
      address: 'PTI, Peremohy 37, Kyiv, UA'
    },
    payment: { card_owner: 'Andrii Rodionov', cardId: 12345678 },
    order_items_id: [
      DBRef("items", ObjectId("61ba1c964f2a97bda1f918d7")),
      DBRef("items", ObjectId("61ba1c964f2a97bda1f918d8"))
    ]
  },
  {
    _id: ObjectId("61ba5c214f2a97bda1f918ed"),
    order_number: 3001,
    date: ISODate("2021-04-11T00:00:00.000Z"),
    total_sum: 1750.7,
    customer: {
      name: 'John',
      surname: 'Doe',
      phones: [ 152221834 ],
      address: 'Somestree, 13, Kyiv, UA'
    },
    payment: { card_owner: 'John Doe', cardId: 123839847 },
    order_items_id: [
      DBRef("items", ObjectId("61ba1c964f2a97bda1f918db")),
      DBRef("items", ObjectId("61ba1c964f2a97bda1f918d9")),
      DBRef("items", ObjectId("61ba1c964f2a97bda1f918dc"))
    ]
  },
  {
    _id: ObjectId("61ba5c214f2a97bda1f918ee"),
    order_number: 3002,
    date: ISODate("2021-05-15T00:00:00.000Z"),
    total_sum: 7365,
    customer: {
      name: 'Erika',
      surname: 'Burg',
      phones: [ 3015222134 ],
      address: 'Strasse, 38a, Kyiv, UA'
    },
    payment: { card_owner: 'Peter Burg', cardId: 2846857 },
    order_items_id: [
      DBRef("items", ObjectId("61ba1c964f2a97bda1f918db")),
      DBRef("items", ObjectId("61ba1c964f2a97bda1f918d8"))
    ]
  },
  {
    _id: ObjectId("61ba5c214f2a97bda1f918ef"),
    order_number: 3003,
    date: ISODate("2021-05-15T00:00:00.000Z"),
    total_sum: 100,
    customer: {
      name: 'Erika',
      surname: 'Burg',
      phones: [ 3015222134 ],
      address: 'Strasse, 38a, Kyiv, UA'
    },
    payment: { card_owner: 'Peter Burg', cardId: 2846857 },
    order_items_id: [
      DBRef("items", ObjectId("61ba1c964f2a97bda1f918da")),
      DBRef("items", ObjectId("61ba1c964f2a97bda1f918d8"))
    ]
  }
]
```

```

{
  _id: ObjectId("61ba5c214f2a97bda1f918f0"),
  order_number: 3003,
  date: ISODate("2021-05-16T00:00:00.000Z"),
  total_sum: 100,
  customer: {
    name: 'Erika',
    surname: 'Burg',
    phones: [ 3015222134 ],
    address: 'Strasse, 38a, Kyiv, UA'
  },
  payment: { card_owner: 'Peter Burg', cardId: 2846857 },
  order_items_id: [
    DBRef("items", ObjectId("61ba1c964f2a97bda1f918da")),
    DBRef("items", ObjectId("61ba1c964f2a97bda1f918d8"))
  ]
}
]
mydb> db.items.find()
[
  {
    _id: ObjectId("61ba1c964f2a97bda1f918d7"),
    category: 'Smartwatch',
    model: 'Mi Band 3',
    producer: 'Xiaomi',
    price: 20
  },
  {
    _id: ObjectId("61ba1c964f2a97bda1f918d8"),
    category: 'Laptop',
    model: 'Uehk737',
    producer: 'Samsung',
    price: 1000
  },
  {
    _id: ObjectId("61ba1c964f2a97bda1f918d9"),
    category: 'Smartweight',
    model: 'Body Scale 2',
    producer: 'Xiaomi',
    price: 100
  },
  {
    _id: ObjectId("61ba1c964f2a97bda1f918db"),
    category: 'Phone',
    model: 'iPhone 6',
    producer: 'Apple',
    price: 600
  },
  {
    _id: ObjectId("61ba1c964f2a97bda1f918dc"),
    category: 'TV',
    model: 'LED 3D 300',
    producer: 'Samsung',
    price: 2000
  }
]

```

Підрахувати скільки одиниць товару є у кожного виробника ("producer")

```

mydb> var mapProducerItems = function() {
...   emit(this.producer, 1);
... };

mydb>

mydb> var reduceSum = function (key, values){
...   return Array.sum(values);
... };

mydb> db.items.mapReduce(
...   mapProducerItems,
...   reduceSum,
...   {out : "items_per_producer"}
... )
DeprecationWarning: Collection.mapReduce() is deprecated. Use an aggregation instead.
See https://docs.mongodb.com/manual/core/map-reduce for details.
{ result: 'items_per_producer', ok: 1 }
mydb> db.items_per_producer.find()
[
  { _id: 'Samsung', value: 3 },
  { _id: 'Xiaomi', value: 2 },
  { _id: 'Apple', value: 1 }
]

```

Підрахувати загальну вартість товарів у кожного виробника ("producer")

```

mydb> var mapProducerPrices = function() {
...   emit(this.producer, this.price);
... };

mydb> db.items.mapReduce(
...   mapProducerPrices,
...   reduceSum,
...   {out : "sum_per_producer"}
... )
{ result: 'sum_per_producer', ok: 1 }
mydb> db.sum_per_producer.find()
[
  { _id: 'Apple', value: 600 },
  { _id: 'Xiaomi', value: 120 },
  { _id: 'Samsung', value: 3600 }
]
mydb>

```

Підрахуйте сумарну вартість замовлень зроблену кожним замовником

```

mydb> var mapCustomerOrderSum = function(){
...     emit(this.customer.name, this.total_sum)
... }

mydb> db.orders.mapReduce(
...     mapCustomerOrderSum,
...     reduceSum,
...     {out : "sum_per_customer"}
... )
{ result: 'sum_per_customer', ok: 1 }
mydb> db.sum_per_customer.find()
[
  { _id: 'Andrii', value: 700.7 },
  { _id: 'Erika', value: 7565 },
  { _id: 'John', value: 1750.7 }
]

```

Підрахуйте сумарну вартість замовлень зроблену кожним замовником за певний період часу (використовуйте query condition)

```

mydb> db.orders.mapReduce(
...     mapCustomerOrderSum,
...     reduceSum,
...     {
...         out : "sum_per_customer_before",
...         query : {date : {$lt: new Date("2021-05-16")}}
...     }
... )
{ result: 'sum_per_customer_before', ok: 1 }
mydb> db.sum_per_
db.sum_per_customer          db.sum_per_customer_before    db.sum_per_customer_2021_04_13  db.sum_per_producer

mydb> db.sum_per_customer_before.find()
[
  { _id: 'John', value: 1750.7 },
  { _id: 'Andrii', value: 700.7 },
  { _id: 'Erika', value: 7465 }
]
mydb>

```

Підрахуйте середню вартість замовлення

```
mydb> var mapOrdersSum = function(){
...   emit("count", {sum: this.total_sum, count: 1})
... }

mydb>

mydb> var reduceAvg = function (key, values){
...   reducedVal = { sum: 0, count: 0 };
...   for (var idx = 0; idx < values.length; idx++) {
...     reducedVal.sum += values[idx].sum;
...     reducedVal.count += values[idx].count;
...   }
...   return reducedVal;
... }

mydb>

mydb> var finalizeAvg = function (key, reducedVal){
...   return reducedVal.sum/reducedVal.count;
... }

mydb>

mydb> db.orders.mapReduce(
...   mapOrdersSum,
...   reduceAvg,
...   {
...     out : "avg_sum_orders",
...     finalize: finalizeAvg
...   }
... )
{ result: 'avg_sum_orders', ok: 1 }
mydb> db.avg_sum_orders.find()
[ { _id: 'count', value: 2003.28000000000002 } ]
```

Підрахуйте середню вартість замовлення кожного покупця

```
mydb> var mapCustomerOrderSumWithCount = function(){
...   emit(this.customer.name, {sum: this.total_sum, count: 1})
... }

mydb>

mydb> db.orders.mapReduce(
...   mapCustomerOrderSumWithCount,
...   reduceAvg,
...   {
...     out : "avg_sum_orders_per_customer",
...     finalize: finalizeAvg
...   }
... )
{ result: 'avg_sum_orders_per_customer', ok: 1 }
mydb> db.avg_sum_orders_per_customer.find()
[
  { _id: 'John', value: 1750.7 },
  { _id: 'Andrii', value: 700.7 },
  { _id: 'Erika', value: 2521.6666666666665 }
]
mydb> □
```


Підрахуйте в скількох замовленнях зустрічався кожен товар (скільки разів він був куплений)

```
mydb> var flatMapItem = function(){
...   for (var idx = 0; idx < this.order_items_id.length; idx++) {
...     var key = this.order_items_id[idx]["$id"];
...     emit(key, 1);
...   }
... }

mydb>

mydb> db.orders.mapReduce(
...   flatMapItem,
...   reduceSum,
...   {out : "item_ordered_times"}
... )
{ result: 'item_ordered_times', ok: 1 }
mydb> db.item_ordered_times.find()
[
  { _id: ObjectId("61ba1c964f2a97bda1f918d9"), value: 1 },
  { _id: ObjectId("61ba1c964f2a97bda1f918d7"), value: 1 },
  { _id: ObjectId("61ba1c964f2a97bda1f918d8"), value: 4 },
  { _id: ObjectId("61ba1c964f2a97bda1f918db"), value: 2 },
  { _id: ObjectId("61ba1c964f2a97bda1f918dc"), value: 1 },
  { _id: ObjectId("61ba1c964f2a97bda1f918da"), value: 2 }
]
mydb> █
```

Для кожного товару отримаєте список всіх замовників які купили його

```
mydb> var reduceToSet = function(key, customers){
...   return [...new Set(customers)];
... }

mydb>

mydb> db.orders.mapReduce(
...   mapItemCustomer,
...   reduceToSet,
...   {out : "item_by_customers_unique"}
... )
{ result: 'item_by_customers_unique', ok: 1 }
mydb> db.item_by_customers_unique.find()
[
  { _id: ObjectId("61ba1c964f2a97bda1f918d9"), value: [ 'John' ] },
  { _id: ObjectId("61ba1c964f2a97bda1f918d7"), value: [ 'Andrii' ] },
  {
    _id: ObjectId("61ba1c964f2a97bda1f918d8"),
    value: [ 'Erika', 'Andrii' ]
  },
  {
    _id: ObjectId("61ba1c964f2a97bda1f918db"),
    value: [ 'Erika', 'John' ]
  },
  { _id: ObjectId("61ba1c964f2a97bda1f918dc"), value: [ 'John' ] },
  { _id: ObjectId("61ba1c964f2a97bda1f918da"), value: [ 'Erika' ] }
]
mydb> █
```

Отримайте товар та список замовників, які купували його більше одного (двох)

```
mydb> var mapItemCustomerCount= function(){
...   for (var idx = 0; idx < this.order_items_id.length; idx++) {
...     var key = this.order_items_id[idx]["$id"];
...     emit({item: key, name:this.customer.name}, 1);
...   }
... }

mydb>

mydb> db.orders.mapReduce(
...   mapItemCustomerCount,
...   reduceSum,
...   {out : "item_by_customers"})
... )
{ result: 'item_by_customers', ok: 1 }
mydb> db.item_by_customers.find()
[
  {
    _id: { item: ObjectId("61ba1c964f2a97bda1f918d9"), name: 'John' },
    value: 1
  },
  {
    _id: { item: ObjectId("61ba1c964f2a97bda1f918dc"), name: 'John' },
    value: 1
  },
  {
    _id: { item: ObjectId("61ba1c964f2a97bda1f918d8"), name: 'Erika' },
    value: 3
  },
  {
    _id: { item: ObjectId("61ba1c964f2a97bda1f918da"), name: 'Erika' },
    value: 2
  },
  {
    _id: { item: ObjectId("61ba1c964f2a97bda1f918db"), name: 'Erika' },
    value: 1
  },
  {
    _id: { item: ObjectId("61ba1c964f2a97bda1f918d7"), name: 'Andrii' },
    value: 1
  },
  {
    _id: { item: ObjectId("61ba1c964f2a97bda1f918d8"), name: 'Andrii' },
    value: 1
  },
  {
    _id: { item: ObjectId("61ba1c964f2a97bda1f918db"), name: 'John' },
    value: 1
  }
]
```

разу(ів)

```

mydb> var mapItemByCustomer = function (){
...     emit(this["_id"].item, this["_id"].name)
... }

mydb>

mydb> var reduceToList = function(key, customers){
...     return customers;
... }

mydb>

mydb> db.item_by_customers.mapReduce(
...     mapItemByCustomer,
...     reduceToList,
...     {
...         out : "item_by_customers_gt_1",
...         query : {value : {$gt:1}}
...     }
... )
{ result: 'item_by_customers_gt_1', ok: 1 }
mydb> db.item_by_customers_gt_1.find()
[
  { _id: ObjectId("61ba1c964f2a97bda1f918d8"), value: [ 'Erika' ] },
  { _id: ObjectId("61ba1c964f2a97bda1f918da"), value: [ 'Erika' ] }
]
mydb>

mydb> db.item_by_customers.mapReduce(
...     mapItemByCustomer,
...     reduceToList,
...     {
...         out : "item_by_customers_gt_2",
...         query : {value : {$gt:2}}
...     }
... )
{ result: 'item_by_customers_gt_2', ok: 1 }
mydb> db.item_by_customers_gt_2.find()
[ { _id: ObjectId("61ba1c964f2a97bda1f918d8"), value: [ 'Erika' ] } ]
mydb>

```


Отримайте топ N товарів за популярністю (тобто топ товарів, які куплялись найчастіше) (функцію sort не застосовувати)

```
mydb> var flatMapItem = function(){
...   for (var idx = 0; idx < this.order_items_id.length; idx++) {
...     var key = this.order_items_id[idx];
...     emit(key, 1);
...   }
... }

mydb>

mydb> var reduceSum = function (key, values){
...   return Array.sum(values);
... };

mydb> db.orders.mapReduce(flatMapItem, reduceSum, {out:'item_ordered_times'});
{ result: 'item_ordered_times', ok: 1 }
mydb>

mydb> db.item_ordered_times.find()
[
  {
    _id: DBRef("items", ObjectId("61ba1c964f2a97bda1f918da")),
    value: 2
  },
  {
    _id: DBRef("items", ObjectId("61ba1c964f2a97bda1f918d8")),
    value: 4
  },
  {
    _id: DBRef("items", ObjectId("61ba1c964f2a97bda1f918dc")),
    value: 1
  },
  {
    _id: DBRef("items", ObjectId("61ba1c964f2a97bda1f918db")),
    value: 2
  },
  {
    _id: DBRef("items", ObjectId("61ba1c964f2a97bda1f918d9")),
    value: 1
  },
  {
    _id: DBRef("items", ObjectId("61ba1c964f2a97bda1f918d7")),
    value: 1
  }
]
```

```

mydb> var mapTimesBought = function(){emit('top', {item: this._id, times: this.value})}

mydb>

mydb> var reduceSort = function(key, values) {
...   let n=3;
...   var topN = [];
...   for (var i = 0; i < n; i++){
...     var max = values[0].times
...     var maxObj = values[0]
...     var index = 0
...     for (var j = 0; j < values.length; j++){
...       if (values[j].times >= max){
...         max = values[j].times;
...         maxObj = values[j];
...         index = j;
...       }
...     }
...     topN.push(maxObj)
...     values.splice(index, 1)
...   }
...   return topN;
... }

mydb> db.item_ordered_times.mapReduce(mapTimesBought, reduceSort,{out:'top_n_items'});
{ result: 'top_n_items', ok: 1 }
mydb> db.top_n_items.find()
[
  {
    _id: 'top',
    value: [
      {
        item: DBRef("items", ObjectId("61ba1c964f2a97bda1f918d8")),
        times: 4
      },
      {
        item: DBRef("items", ObjectId("61ba1c964f2a97bda1f918da")),
        times: 2
      },
      {
        item: DBRef("items", ObjectId("61ba1c964f2a97bda1f918db")),
        times: 2
      }
    ]
  }
]
mydb>

```

Для завдання 4) реалізуйте [інкрементальний](#) Map / Reduce використовуючи *out i action*

```
mydb> db.sum_per_customer_before.find()
[
  { _id: 'Andrii', value: 700.7 },
  { _id: 'Erika', value: 7465 },
  { _id: 'John', value: 1750.7 }
]
mydb> db.orders.insertOne({
...   "order_number": 3005, "date": ISODate("2021-05-17"), "total_sum": 9999,
...   "customer": { "name": "Andrii", "surname": "Rodionov", "phones": [9876543, 1234567], "address": "PTI, Peremohy 37, Kyiv, UA"},
...   "payment": { "card_owner": "Andrii Rodionov", "cardId": 12345678},
...   "order_items_id": [{"Sref": "items", "$id": ObjectId("61ba1c964f2a97bda1f918da")}]}
... })
{
  acknowledged: true,
  insertedId: ObjectId("61ba76dc4f2a97bda1f918f7")
}
mydb> db.orders.mapReduce(
...   mapCustomerOrderSum,
...   reduceSum,
...   {
...     out: {reduce: "sum_per_customer_before"},
...     query: {date: {$gt: new Date("2021-05-16")}}
...   }
... )
{ result: 'sum_per_customer_before', ok: 1 }
mydb> db.sum_per_customer_before.find()
[
  { _id: 'Andrii', value: 10699.7 },
  { _id: 'Erika', value: 7465 },
  { _id: 'John', value: 1750.7 }
]
```

Для кожного користувача, визначити на яку суму їм було зроблено замовлень за кожен місяць цього року та за аналогічний місяць минулого року та динаміку збільшення/зменшення замовлень.

Спершу, добавимо декілька замовлень до вже існуючих, які були створені у цьому та минулому році

```
b.orders.insertMany([
...   {
...     "order_number": 3005, "date": ISODate("2020-05-12"), "total_sum": 6000,
...     "customer": { "name": "Andrii", "surname": "Rodionov", "phones": [9876543, 1234567], "address": "PTI, Peremohy 37, Kyiv, UA"},
...     "payment": { "card_owner": "Andrii Rodionov", "cardId": 12345678},
...     "order_items_id": [{"Sref": "items", "$id": ObjectId("61ba1c964f2a97bda1f918da")}]}
...   },
...   {
...     "order_number": 3006, "date": ISODate("2021-05-12"), "total_sum": 3000,
...     "customer": { "name": "Andrii", "surname": "Rodionov", "phones": [9876543, 1234567], "address": "PTI, Peremohy 37, Kyiv, UA"},
...     "payment": { "card_owner": "Andrii Rodionov", "cardId": 12345678},
...     "order_items_id": [{"Sref": "items", "$id": ObjectId("61ba1c964f2a97bda1f918da")}]}
...   },
...   {
...     "order_number": 3007, "date": ISODate("2020-04-12"), "total_sum": 7000,
...     "customer": { "name": "Andrii", "surname": "Rodionov", "phones": [9876543, 1234567], "address": "PTI, Peremohy 37, Kyiv, UA"},
...     "payment": { "card_owner": "Andrii Rodionov", "cardId": 12345678},
...     "order_items_id": [{"Sref": "items", "$id": ObjectId("61ba1c964f2a97bda1f918da")}]}
...   },
...   {
...     "order_number": 3008, "date": ISODate("2020-05-16"), "total_sum": 100,
...     "customer": { "name": "Erika", "surname": "Burg", "phones": [3015222134], "address": "Strasse, 38a, Kyiv, UA"},
...     "payment": { "card_owner": "Peter Burg", "cardId": 2846857},
...     "order_items_id": [{"Sref": "items", "$id": ObjectId("61ba1c964f2a97bda1f918da")}]}
...   },
...   {
...     "order_number": 3009, "date": ISODate("2021-04-01"), "total_sum": 120,
...     "customer": { "name": "Erika", "surname": "Burg", "phones": [3015222134], "address": "Strasse, 38a, Kyiv, UA"},
...     "payment": { "card_owner": "Peter Burg", "cardId": 2846857},
...     "order_items_id": [{"Sref": "items", "$id": ObjectId("61ba1c964f2a97bda1f918da")}]}
...   },
...   {
...     "order_number": 3009, "date": ISODate("2021-04-16"), "total_sum": 130,
...     "customer": { "name": "Erika", "surname": "Burg", "phones": [3015222134], "address": "Strasse, 38a, Kyiv, UA"},
...     "payment": { "card_owner": "Peter Burg", "cardId": 2846857},
...     "order_items_id": [{"Sref": "items", "$id": ObjectId("61ba1c964f2a97bda1f918da")}]}
...   }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("61bb60d34f2a97bda1f9190e"),
    '1': ObjectId("61bb60d34f2a97bda1f9190f"),
    '2': ObjectId("61bb60d34f2a97bda1f91910"),
    '3': ObjectId("61bb60d34f2a97bda1f91911"),
    '4': ObjectId("61bb60d34f2a97bda1f91912"),
    '5': ObjectId("61bb60d34f2a97bda1f91913")
  }
}
mydb>
```

```

mydb> var mapFullInfo = function(){
...   var curr_year = new Date().getFullYear();
...   emit(
...     {name: this.customer.name, month: this.date.getMonth() + 1},
...     {
...       name : this.customer.name,
...       year: this.date.getFullYear(),
...       month: this.date.getMonth() + 1,
...       amount: this.date.getFullYear() == curr_year ? this.total_sum : 0,
...       prev_year_amount: this.date.getFullYear() == curr_year - 1 ? this.total_sum : 0
...     })
... }

mydb>

mydb> var reduceFullInfo = function(key, values){
...   var curr_year = new Date().getFullYear();
...   var reducedObj = {
...     name : key.name,
...     year: curr_year,
...     month: key.month,
...     amount: 0,
...     prev_year_amount: 0
...   }
...   for (var idx = 0; idx < values.length; idx++) {
...     var obj = values[idx];
...     if (parseInt(obj.year) == curr_year){
...       reducedObj.amount += obj.amount;
...     }
...     if (parseInt(obj.year) == curr_year-1){
...       reducedObj.prev_year_amount += obj.prev_year_amount;
...     }
...   }
...   return reducedObj;
... }

mydb>

mydb> var finalizeFullInfo = function (key, reducedObj){
...   reducedObj.diff = reducedObj.amount - reducedObj.prev_year_amount;
...   return reducedObj;
... }

mydb>

```

```

mydb> db.orders.mapReduce(
...   mapFullInfo,
...   reduceFullInfo,
...   {
...     out : "full_info",
...     finalize: finalizeFullInfo
...   }
... )
{ result: 'full_info', ok: 1 }

```

```
{
  _id: { name: 'Andrii', month: 4 },
  value: {
    name: 'Andrii',
    year: 2021,
    month: 4,
    amount: 700.7,
    prev_year_amount: 7000,
    diff: -6299.3
  }
},
{
  _id: { name: 'Andrii', month: 5 },
  value: {
    name: 'Andrii',
    year: 2021,
    month: 5,
    amount: 3000,
    prev_year_amount: 6000,
    diff: -3000
  }
},
{
  _id: { name: 'John', month: 4 },
  value: {
    name: 'John',
    year: 2021,
    month: 4,
    amount: 1750.7,
    prev_year_amount: 0,
    diff: 1750.7
  }
},
{
  _id: { name: 'Erika', month: 5 },
  value: {
    name: 'Erika',
    year: 2021,
    month: 5,
    amount: 7565,
    prev_year_amount: 100,
    diff: 7465
  }
},
{
  _id: { name: 'Erika', month: 4 },
  value: {
    name: 'Erika',
    year: 2021,
    month: 4,
    amount: 250,
    prev_year_amount: 0,
    diff: 250
  }
}
}
```