

- Використовуйте дані та колекції з завдання 1.

```
> db.items.find()
{ "_id" : ObjectId("60bf4d5e9dfbbf2b90a17552"), "category" : "Phone", "model" : "iPhone 6", "producer" : "Apple", "price" : 600 }
{ "_id" : ObjectId("60bf4d5e9dfbbf2b90a17553"), "category" : "Phone", "model" : "iPhone 12", "producer" : "Apple", "price" : 700 }
{ "_id" : ObjectId("60bf4d5e9dfbbf2b90a17554"), "category" : "TV", "model" : "MySuperTV Model 6s", "producer" : "SamSong", "outOfStock" : true }
{ "_id" : ObjectId("60bf4d5e9dfbbf2b90a17555"), "category" : "Smart Watch", "model" : "SamSong Galactic Watch", "producer" : "SamSong", "price" : 400 }
{ "_id" : ObjectId("60bf4d5e9dfbbf2b90a17556"), "category" : "Smart Watch", "model" : "Peach Watch 2", "producer" : "Peach", "price" : 300 }
{ "_id" : ObjectId("60bf4d5e9dfbbf2b90a17557"), "category" : "Charger", "model" : "iCharger", "producer" : "Apple", "price" : 700 }
> db.orders.find()
{ "_id" : ObjectId("60bf4d5f9dfbbf2b90a17558"), "order_number" : 1, "date" : ISODate("2015-04-14T00:00:00Z"), "total_sum" : 1300, "customer" : { "name" : "Andrii", "surname" : "Rodinov", "phones" : [ 9876543, 1234567 ], "address" : "PTI, Peremohy 37, Kyiv, UA" }, "payment" : { "card_owner" : "Andrii Rodionov", "cardId" : 12345678 }, "order_items_id" : [ DBRef("items", ObjectId("60bf4d5e9dfbbf2b90a17552")), DBRef("items", ObjectId("60bf4d5e9dfbbf2b90a17553")) ] }
{ "_id" : ObjectId("60bf4d5f9dfbbf2b90a17559"), "order_number" : 2, "date" : ISODate("2018-04-14T00:00:00Z"), "total_sum" : 600, "customer" : { "name" : "Neandrii", "surname" : "Nerodionov", "phones" : [ 9875644, 1234568 ], "address" : "PTI, Peremohy 38, Kyiv, UA" }, "payment" : { "card_owner" : "Neandrii Nerodionov", "cardId" : 12345679 }, "order_items_id" : [ DBRef("items", ObjectId("60bf4d5e9dfbbf2b90a17552")), DBRef("items", ObjectId("60bf4d5e9dfbbf2b90a17554")) ] }
> []
```

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

```
> db.items.mapReduce(
...   function() { emit(this.producer, 1); },
...   function(k, v) { return Array.sum(v); },
...   { out: "count" },
... );
{ "result" : "count", "ok" : 1 }
> db.count.find()
{ "_id" : "Apple", "value" : 3 }
{ "_id" : "SamSong", "value" : 2 }
{ "_id" : "Peach", "value" : 1 }
> []
```

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

```

> db.items.mapReduce(
...   function() { emit(this.producer, this.price || 0); },
...   function(k, v) { return Array.sum(v); },
...   { out: "producer_sum" },
... );
{ "result" : "producer_sum", "ok" : 1 }
> db.producer_sum.find()
{ "_id" : "Apple", "value" : 2000 }
{ "_id" : "SamSong", "value" : 400 }
{ "_id" : "Peach", "value" : 300 }
> 

```

Підрахуйте сумарну вартість замовлень зроблену кожним замовником

```

> db.orders.mapReduce(
...   function() { emit(this.customer.name + " " + this.customer.surname, this.total_sum || 0); },
...   function(k, v) { return Array.sum(v); },
...   { out: "customer_sum" },
... );
{ "result" : "customer_sum", "ok" : 1 }
>
> db.customer_sum.find()
{ "_id" : "Andrii Rodinov", "value" : 1300 }
{ "_id" : "Neandrii Nerodinov", "value" : 600 }
> 

```

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

```

> db.orders.mapReduce(
...   function() { emit(this.customer.name + " " + this.customer.surname, this.total_sum || 0); },
...   function(k, v) { return Array.sum(v); },
...   { query: { date: { $gt: ISODate("2015-05-14") } }, out: "customer_sum_gt" },
... );
{ "result" : "customer_sum_gt", "ok" : 1 }
> db.customer_sum_gt.find()
{ "_id" : "Neandrii Nerodinov", "value" : 600 }
> 

```

Підрахуйте середню вартість замовлення

```

> db.orders.mapReduce(
...   function() { emit(1, this.total_sum || 0); },
...   function(k, v) { return Array.avg(v); },
...   { out: "avg_order_sum" },
... );
{ "result" : "avg_order_sum", "ok" : 1 }
> db.avg_order_sum.find()
{ "_id" : 1, "value" : 950 }
> 

```

Підрахуйте середню вартість замовлення кожного покупця

```

> db.orders.mapReduce(
...   function() { emit(this.customer.name + " " + this.customer.surname, this.total_sum || 0); },
...   function(k, v) { return Array.avg(v); },
...   { out: "customer_avg_sum" },
... );
{ "result" : "customer_avg_sum", "ok" : 1 }
> db.customer_avg_sum.find()
{ "_id" : "Andrii Rodinov", "value" : 1300 }
{ "_id" : "Neandrii Nerodinov", "value" : 600 }
> 

```

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

```

> db.orders.mapReduce(
...   function() { this.order_items_id.forEach((i) => emit(i, 1)) },
...   function(k, v) { return Array.sum(v); },
...   { out: "item_order_count" },
... );
{ "result" : "item_order_count", "ok" : 1 }
> db.item_order_count.find()
{ "_id" : DBRef("items", ObjectId("60bf4d5e9dfbbf2b90a17554")), "value" : 1 }
{ "_id" : DBRef("items", ObjectId("60bf4d5e9dfbbf2b90a17553")), "value" : 1 }
{ "_id" : DBRef("items", ObjectId("60bf4d5e9dfbbf2b90a17552")), "value" : 2 }
> 

```

Для кожного товару отримаєте список всіх замовників які купили його

```

> db.orders.mapReduce(
...   function() { this.order_items_id.forEach((i) => emit(i, this.customer.name + " " + this.customer.surname)) },
...   function(k, v) { print(v); return Array.from(new Set(v)); },
...   { out: "item_customers" },
... );
{ "result" : "item_customers", "ok" : 1 }
> db.item_customers.find()
{ "_id" : DBRef("items", ObjectId("60bf4d5e9dfbbf2b90a17554")), "value" : [ "Neandrii Nerodinov" ] }
{ "_id" : DBRef("items", ObjectId("60bf4d5e9dfbbf2b90a17553")), "value" : [ "Andrii Rodinov" ] }
{ "_id" : DBRef("items", ObjectId("60bf4d5e9dfbbf2b90a17552")), "value" : [ "Neandrii Nerodinov", "Andrii Rodinov" ] }
> 

```

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

```

> db.orders.mapReduce(
...   function() { this.order_items_id.forEach((i) => emit(i, this.customer.name + " " + this.customer.surname)) },
...   function(k, v) {
...     let m = {};
...     v.forEach((val) => {
...       m[val] = (m[val] || 0) + 1;
...     });
...     return Object.entries(m).filter(([k, v]) => v > 1).map(([k, v]) => k);
...   },
...   { out: "item_bought_more_than_once" },
... );
{ "result" : "item_bought_more_than_once", "ok" : 1 }
> db.item_bought_more_than_once.find()
{ "_id" : DBRef("items", ObjectId("60bf4d5e9dfbbf2b90a17552")), "value" : [ ] }
{ "_id" : DBRef("items", ObjectId("60bf4d5e9dfbbf2b90a17553")), "value" : [ ] }
{ "_id" : DBRef("items", ObjectId("60bf4d5e9dfbbf2b90a17554")), "value" : [ ] }
> // no such items, I've checked

```

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

```

> db.orders.mapReduce(
...   function() { emit(this.customer.name + " " + this.customer.surname, this.total_sum || 0); },
...   function(k, v) { return Array.sum(v); },
...   // split data
...   { query: { date: {$gt: ISODate("2015-05-14"), $lte: ISODate("2016-05-15")}}, out: "tmp" },
... );
{ "result" : "tmp", "ok" : 1 }
> db.tmp.find()
> db.orders.mapReduce(
...   function() { emit(this.customer.name + " " + this.customer.surname, this.total_sum || 0); },
...   function(k, v) { return Array.sum(v); },
...   { query: { date: {$gt: ISODate("2016-05-15")}}, out: {reduce: "tmp"} },
... );
{ "result" : "tmp", "ok" : 1 }
> db.tmp.find()
{ "_id" : "Neandrii Nerodinov", "value" : 600 }
// we have only two orders ~\_('▽)~/~
> 

```

Підрахуємо для всіх замовлень, щоб хоч якось продемонструвати роботу інкрементального мапредьюсу

```

> db.tmp.remove({})
WriteResult({ "nRemoved" : 1 })
> db.orders.mapReduce(
...   function() { emit(this.customer.name + " " + this.customer.surname, this.total_sum || 0); },
...   function(k, v) { return Array.sum(v); },
...   { query: { date: {$lte: ISODate("2016-05-15")}}, out: "tmp" },
... );
{ "result" : "tmp", "ok" : 1 }
> db.tmp.find()
{ "_id" : "Andrii Rodinov", "value" : 1300 }
> db.orders.mapReduce(
...   function() { emit(this.customer.name + " " + this.customer.surname, this.total_sum || 0); },
...   function(k, v) { return Array.sum(v); },
...   { query: { date: {$gt: ISODate("2016-05-15")}}, out: {reduce: "tmp"} },
... );
{ "result" : "tmp", "ok" : 1 }
> db.tmp.find()
{ "_id" : "Andrii Rodinov", "value" : 1300 }
{ "_id" : "Neandrii Nerodinov", "value" : 600 }
> 

```

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

```

> db.orders.mapReduce(
...   function() {
...     let year = this.date.getFullYear();
...     emit(
...       this.customer.name + " " + this.customer.surname,
...       {
...         month: this.date.getMonth(),
...         year,
...         previous_year: year - 1,
...         total_sum: this.total_sum
...       }
...     );
...   },
...   function (k, v) {
...     let values = v.sort((a, b) => b - a);
...     let sum = values.reduce((s, i) => s + i.total_sum, 0);
...     let first = values[0];
...     let sum2 = values
...       .filter((i) => i.year === first.previous_year && i.month === first.month)
...       .reduce((s, i) => s + i.total_sum, 0);
...     let diff = sum - sum2;
...     let diff_prefix = "";
...     if (diff > 0) {
...       diff_prefix = "+";
...     } else {
...       diff_prefix = "-";
...     }
...     let y = values.reduce((s, i) => i.year > s ? i.year : s, -Infinity);
...     return {
...       customer: k,
...       month: first.month + 1,
...       year: y,
...       total_sum: sum,
...       prev_year_amount: sum2,
...       diff: diff_prefix + diff.toString(),
...     };
...   },
...   {out: {inline: 1}}
... )

```

```
{
  "results" : [
    {
      "_id" : "Andrii Rodinov",
      "value" : {
        "customer" : "Andrii Rodinov",
        "month" : 4,
        "year" : 2015,
        "total_sum" : 1300,
        "prev_year_amount" : 0,
        "diff" : "+1300"
      }
    },
    {
      "_id" : "Neandrii Nerodinov",
      "value" : {
        "customer" : "Neandrii Nerodinov",
        "month" : 4,
        "year" : 2018,
        "total_sum" : 600,
        "prev_year_amount" : 0,
        "diff" : "+600"
      }
    }
  ],
  "ok" : 1
}
```

> 