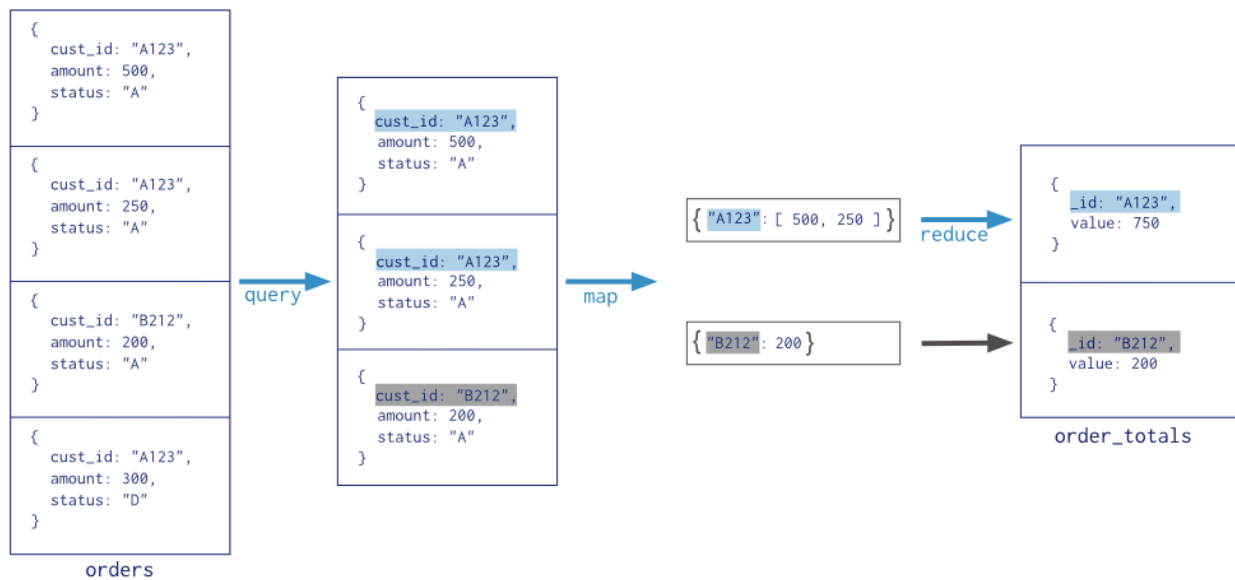# Map Reduce

```
Collection
  │
  ▼
db.orders.mapReduce(
        map    ──────▶   function() { emit( this.cust_id, this.amount ); },
        reduce ──────▶   function(key, values) { return Array.sum( values ) },
                         {
        query  ──────▶     query: { status: "A" },
        output ──────▶     out: "order_totals"
                         }
                      )
```

```
{
  cust_id: "A123",
  amount: 500,
  status: "A"
}

{
  cust_id: "A123",
  amount: 250,
  status: "A"
}

{
  cust_id: "B212",
  amount: 200,
  status: "A"
}

{
  cust_id: "A123",
  amount: 300,
  status: "D"
}
```
orders

query ──▶

```
{
  cust_id: "A123",
  amount: 500,
  status: "A"
}

{
  cust_id: "A123",
  amount: 250,
  status: "A"
}

{
  cust_id: "B212",
  amount: 200,
  status: "A"
}
```

map ──▶

```
{ "A123": [ 500, 250 ] }

{ "B212": 200 }
```

reduce ──▶

```
{
  _id: "A123",
  value: 750
}

{
  _id: "B212",
  value: 200
}
```
order_totals

> **use customer**
switched to db customer

> **db.orders.insert({cust_id:"A123", Amount:500, status:"A"})**
> **db.orders.insert({cust_id:"A123", Amount:250, status:"A"})**
> **db.orders.insert({cust_id:"B212", Amount:200, status:"A"})**
> **db.orders.insert({cust_id:"B212", Amount:300, status:"D"})**
> **db.orders.insert({cust_id:"A123", Amount:300, status:"D"})**

> **db.orders.find()**
{ "_id" : ObjectId("54184428541fb1f5e350f06d"), "cust_id" : "A123", "Amount" : 500, "status" : "A" }

{ "_id" : ObjectId("5418467c541fb1f5e350f06e"), "cust_id" : "A123", "Amount" : 250, "status" : "A" }

{ "_id" : ObjectId("541846c6541fb1f5e350f06f"), "cust_id" : "B212", "Amount" : 200, "status" : "A" }

{ "_id" : ObjectId("541846e1541fb1f5e350f070"), "cust_id" : "B212", "Amount" : 300, "status" : "D" }

{ "_id" : ObjectId("54184713541fb1f5e350f071"), "cust_id" : "A123", "Amount" : 300, "status" : "D" }

Map

```
var map = function(){
  emit(this.cust_id, this.Amount);
};
```

Reduce

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

MapReduce

```
> db.orders.mapReduce(
        map,
        reduce,
        {
                query:{status:"A"},
                out:"order_totals"
        }
);
```
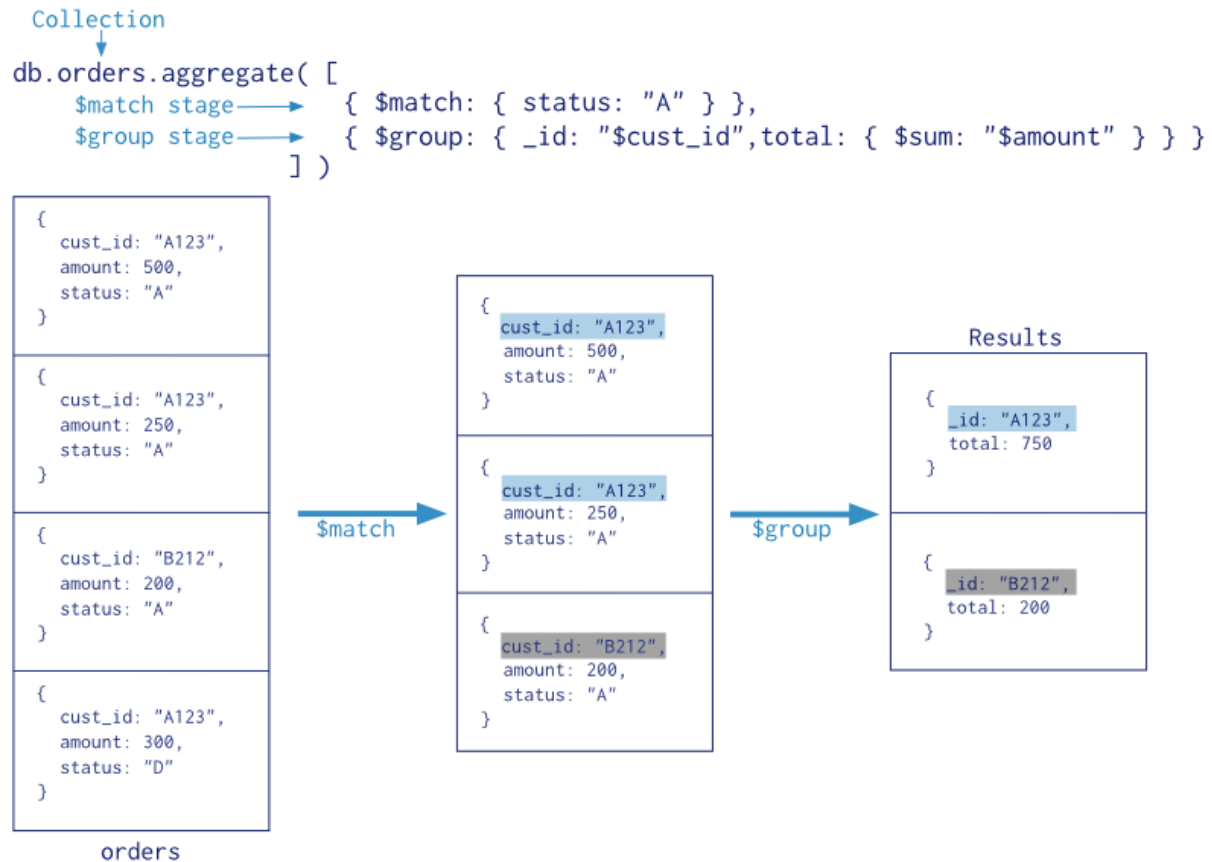
Résultats

```
{
    "result" : "order_totals",
    "timeMillis" : 31,
    "counts" : {
        "input" : 3,
        "emit" : 3,
        "reduce" : 1,
        "output" : 2
    },
    "ok" : 1,
}

> db.order_totals.find()
{ "_id" : "A123", "value" : 750 }
{ "_id" : "B212", "value" : 200 }
```

Aggregation Framework

Nous allons faire la même chose avec l'aggrégation framework.

> **db.orders.aggregate([{$match:{status:"A"}},{$group:{_id:"$cust_id",total:{$sum:"$Amount"}}}])**
{ "_id" : "B212", "total" : 200 }
{ "_id" : "A123", "total" : 750 }