

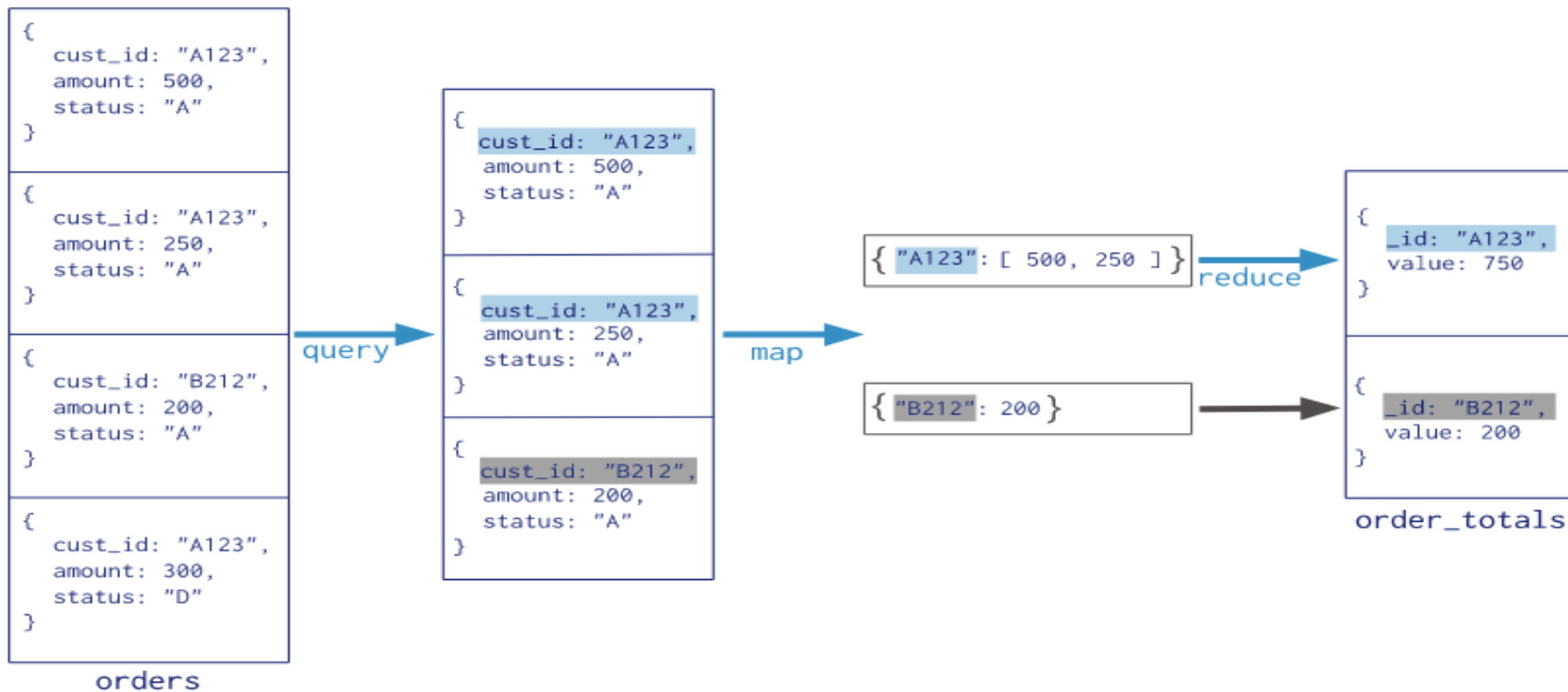
MAP REDUCE

- mapReduce es un sistema de agregación diseñado para manejar cantidades enormes de datos (petabytes), por lo que suele ejecutarse en sistemas de procesamiento en paralelo, como Hadoop. No obstante, MongoDB también incluye una implementación de mapReduce.
- Su funcionamiento básico consiste en utilizar 2 funciones escritas en JavaScript:
 - map: Esta función extrae de la colección pares clave: valor. La clave debe ser un campo de la colección. El valor será un array con los valores de todos los documentos que comparten la misma clave.
 - reduce: Esta función es la que realmente agrega los valores de los pares anteriores actuando sobre el array de valores.
- Adicionalmente admite otros parámetros, como:
 - finalize: Se aplica a continuación de reduce para alterar la presentación de los datos.
 - out: Permite especificar el destino del resultado de la operación mapReduce: la consola o una colección.
 - query: Sirve para limitar el número de documentos que debe procesar la función map.

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"
})



MAP REDUCE

MUY IMPORTANTE

El tipo de dato devuelto por *reduce* tiene que ser idéntico al tipo del valor emitido por la función map.

Es posible que la función *reduce* se ejecute **varias veces para una misma clave** (en tal caso la reducción anterior se toma como un valor de entrada más para la reducción actual)

MAP REDUCE

Ejemplo:

- Vamos a calcular la población total de cada ciudad de nuestra colección de zips (códigos postales)

```
var funcion_map=function(){emit(this.city,this.pop);}  
var funcion_reduce=function(ciudad,poblacion){return Array.sum(poblacion);}  
db.zips.mapReduce(funcion_map,funcion_reduce,{out:{inline:1}})  
  
...  
  
{  
  "_id" : "ZWOLLE", "value" : 5325  
}  
  
],  
  
"timeMillis" : 2381,  
"counts" : {"input" : 29353,"emit" : 29353,"reduce" : 8386,"output" : 16584},  
"ok" : 1  
}
```

MAP REDUCE

¿Y si quisiéramos pasar el nombre del estado?

Usamos el "truco" de emitir como valor un documento JSON que incluya los campos que nos interesen

```
var funcion_map=function(){  
    var estado_poblacion={estado: this.state, poblacion: this.pop};  
    emit(this.city,estado_poblacion);  
}  
  
var funcion_reduce=function(ciudad,estado_poblacion){  
    var resultado_reduce={ciudad:"",poblacion:0};  
    for(var i=0; i<estado_poblacion.length; i++){  
        resultado_reduce.poblacion+=estado_poblacion[i].poblacion;  
    }  
    resultado_reduce.ciudad=estado_poblacion[0].estado;  
    return resultado_reduce;  
}  
  
db.zips.mapReduce(funcion_map,funcion_reduce,{out:{inline:1}})
```

MAP REDUCE

Vamos a calcular la población media de las ciudades de cada estado

Planteamiento

- 1) Hacemos un primer *mapReduce* con una clave compuesta por el estado y la ciudad, para conseguir reducir la población total de cada ciudad sumando la de cada uno de sus zips y lo almacenamos en una colección llamada *zips_ageregados_por_ciudad*.
- 2) Realizamos un segundo mapReduce usando como clave el estado de la colección generada en el paso anterior, y acumulando en un contador el número de ciudades que tiene cada estado.

MAP REDUCE

Paso 1

```
var funcion_map=function() {  
    emit({estado:this.state,ciudad:this.city},this.pop);  
}  
  
var funcion_reduce=function(estado_ciudad,poblacion) {  
    return Array.sum(poblacion);  
}  
  
db.zips.mapReduce(funcion_map,funcion_reduce,{out:"zip_agregados_por_ciudad"})
```

MAP REDUCE

Paso 2

```
var funcion_map2=function(){  
    emit(this._id.estado,{poblacion: this.value, cuenta:1});  
}  
  
var funcion_reduce2=function(estado,poblacion){  
    respuesta_reduce={poblacion: 0, cuenta:0};  
    for(var i=0;i<poblacion.length;i++){  
        respuesta_reduce.poblacion+=poblacion[i].poblacion;  
        respuesta_reduce.cuenta+=poblacion[i].cuenta;  
    }  
    return respuesta_reduce;  
}  
  
funcion_finalize=function(estado,poblacion){  
    poblacion.media=poblacion.poblacion/poblacion.cuenta;  
    return poblacion;  
}  
  
db.zips_agregados_por_ciudad.mapReduce(funcion_map2,funcion_reduce2,{out:inline:1},  
    finalize:funcion_finalize)
```