

# Exemple 1

```
> mongo
```

```
> use mapreduce
```

```
switched to db mapreduce
```

Construire une collection de commande avec la structure suivante :

- **userid** : l'identifiant de l'utilisateur à l'origine de la commande
- **date** : la date de passage (ou de validation, pourquoi pas) de la commande
- **codepost** : le code postal de la ville de résidence de l'utilisateur
- **articles** : la liste des articles commandés (il y en a autant que souhaité)
- **totalttc** : le prix TTC de la totalité de la commande
- **totalht** : le prix HT de la totalité de la commande
- **tva** : le montant de TVA applicable à la commande

```
db.commandes.insert({userid: 54845, date: new Date("Apr 28, 2013"), codepost:13100, articles:[{id:1,nom:'livre',prix:29.90}, {id:9, nom:'eponge', prix:2.90}], totalttc:32.80, tva:5.38, totalht:27.42});
```

```
db.commandes.insert({userid: 54846, date: new Date("Apr 29, 2013"),codepost:13290, articles:[{id:45,nom:'robinet',prix:69.90}, {id:9, nom:'laitx6', prix:9.90}], totalttc:79.80, tva:13.08, totalht:66.72});
```

```
db.commandes.insert({userid: 54847, date: new Date("Apr 30, 2013"),codepost:13008, articles:[{id:76,nom:'clavier',prix:49.90}, {id:2, nom:'fromage', prix:1.50}], totalttc:51.40, tva:8.42, totalht:42.98});
```

```
db.commandes.insert({userid: 54848, date: new Date("Apr 28, 2013"),codepost:13600, articles:[{id:2987,nom:'presse',prix:2}], totalttc:2, tva:0.33, totalht:1.67});
```

```
db.commandes.insert({userid: 54848, date: new Date("Apr 29, 2013"),codepost:13600, articles:[{id:2988,nom:'presse',prix:5.90}], totalttc:5.90, tva:0.97, totalht:4.93});
```

```
db.commandes.insert({userid: 54848, date: new Date("Apr 30, 2013"),codepost:13600, articles:[{id:3989,nom:'presse',prix:1.20}], totalttc:1.20, tva:0.20, totalht:1});
```

```
db.commandes.insert({userid: 54847, date: new Date("Apr 25, 2013"),codepost:13008, articles:[{id:2987,nom:'presse',prix:2}], totalttc:2, tva:0.33, totalht:1.67});
```

## Structure du JSON

```
{
  "_id": ObjectId("517fb463b53bb7169584f3c7"),
  "userid": 54847,
  "date": ISODate("2013-04-29T22:00:00Z"),
  "codepost": 13008,
  "articles": [
    {
      "id": 76,
      "nom": "clavier",
      "prix": 49.9
    },
    {
      "id": 2,
      "nom": "fromage",
      "prix": 1.5
    }
  ],
  "totalttc": 51.4,
  "tva": 8.42,
  "totalht": 42.98
}
```

## Le JavaScript de notre map-reduce

Notre but est d'afficher, par code postal, le chiffre d'affaire généré par notre site en ligne.

```
var map = function () {
  emit(this.codepost, {totalttc:this.totalttc});
}
var reduce = function (cle, valeur) {
  var s = {somme:0};
  valeur.forEach(function (article) {s.somme += article.totalttc;});
  return s;
}
```

```
db.commandes.mapReduce(map, reduce, {out:'total_cmdes_par_ville'});
```

```
db.total_cmdes_par_ville.find();
```

```
{ "_id" : 13008, "value" : { "somme" : 53.4 } }
{ "_id" : 13100, "value" : { "totalttc" : 32.8 } }
{ "_id" : 13290, "value" : { "totalttc" : 79.8 } }
{ "_id" : 13600, "value" : { "somme" : 9.1 } }
```

## Exemple 2

Pour aller plus loin voir le site de mongodb, dans le tutorial → aggregation exemple

<http://docs.mongodb.org/manual/tutorial/aggregation-with-user-preference-data/>

The screenshot shows the MongoDB Manual website. The left sidebar contains a navigation menu with the following items: MONGODB MANUAL 2.6 (current), Installation, MongoDB CRUD Operations, Data Models, Administration, Security, Aggregation (with sub-items: Aggregation Introduction, Aggregation Concepts, Aggregation Examples, Aggregation with the Zip Code Data Set, Aggregation with User Preference Data, Map-Reduce Examples, Perform Incremental Map-Reduce, Troubleshoot the Map Function, Troubleshoot the Reduce Function, Aggregation Reference), and Indexes. The 'Aggregation Examples' item is highlighted with a red box. The main content area is titled 'Aggregation > Aggregation Examples > Aggregation with User Preference Data'. It features a 'Data Model' section with a JSON example of user data:

```
{
  "_id": "jane",
  "joined": ISODate("2011-03-02"),
  "likes": ["golf", "racquetball"]
}
{
  "_id": "joe",
  "joined": ISODate("2012-07-02"),
  "likes": ["tennis", "golf", "swimming"]
}
```

Below the JSON example is the 'Normalize and Sort Documents' section, which states: 'The following operation returns user names in upper case and in alphabetical order. The aggregation includes user names for all documents in the users collection. You might do this to normalize user names for processing.'

On the right side of the page, there is a 'ON THIS PAGE' section with links to: Data Model, Normalize and Sort Documents, Return Usernames Ordered by Join Month, Return Total Number of Joins per Month, and Return the Five Most Common 'Likes'.