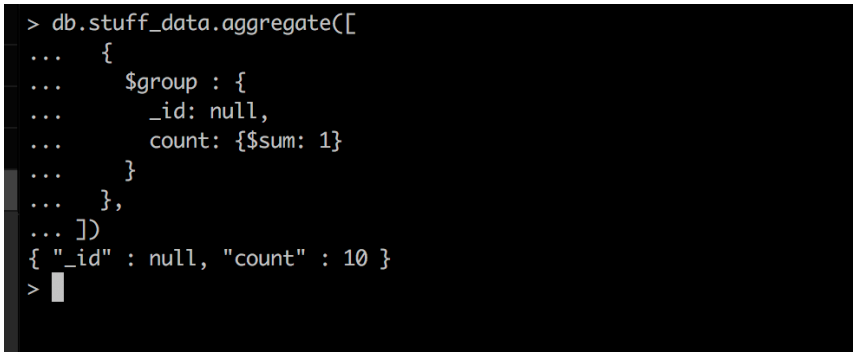
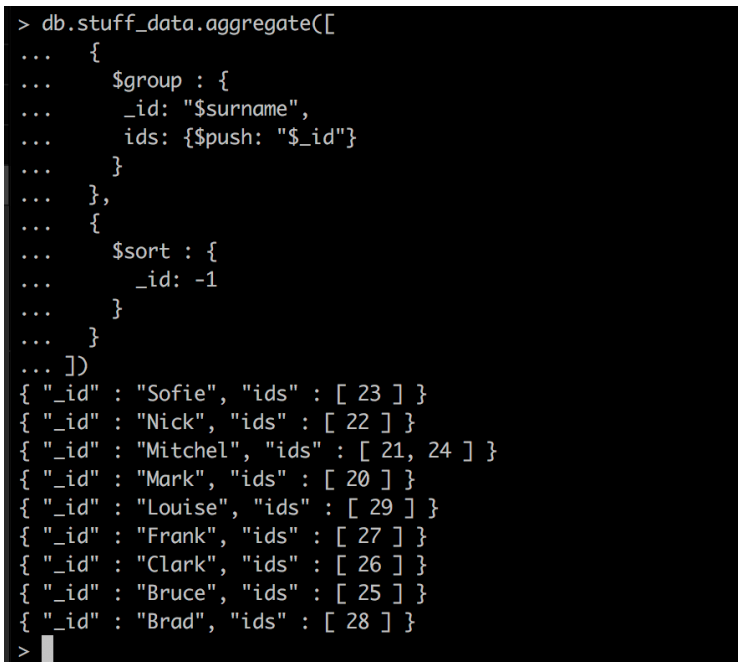
```
-- generate_data python script
import json
import numpy as np

NAMES = ['Brad', 'Mitchel', 'Sofie', 'Clark', 'Bruce', 'David',
'Mark', 'Frank', 'Nick', 'Louise', 'James']

BRANDS = ['Volkswagen', 'Ferrari', 'Ford', 'Volvo', 'Schkoda', 'Suzuki', 'Subaru', 'Mercedes', 'BMW']

TYPES_CODE = ['Small fixes', 'Middle fixes', 'Large fixes', 'Repair wheels', 'Recolor', 'Improve handle', 'Paint wheels',
'Paint base', 'Inside cleaning',
'Outside cleaning', 'Full cleaning']

if __name__ == '__main__':
    with open('trailers_park_data.json', 'w')  as opened:
        with open('stuff_data.json', 'w') as opened_2:
            with open('repairs_data.json', 'w') as opened_3:
                for i in range(10):
                    index_brand = np.random.randint(0, len(BRANDS))
                    auto_num = np.random.randint(1000, 9999)
                    data = {
                        "_id": i,
                        "auto_num": auto_num,
                        "auto_brand": BRANDS[index_brand],
                        "released_date": str(np.random.randint(1, 30)) + "." + str(np.random.randint(1, 12)) + "." +
                        str(np.random.randint(1, 99)),
                        "garage_number": np.random.randint(1, 1000),
                    }
                    json.dump(data, opened, sort_keys=True, indent=4)
                    opened.write(',')

                    index_name = np.random.randint(0, len(NAMES))
                    user_data = {
                        "_id": i + 20,
                        "surname": NAMES[index_name],
                        "ipb": "A.A",
                        "garage_number_id": i,
                        "start_date": str(np.random.randint(1, 30)) + "." + str(np.random.randint(1, 12)) + "." +
                        str(np.random.randint(1, 99)),
                        "end_date": str(np.random.randint(1, 30)) + "." + str(np.random.randint(1, 12)) + "." +
                        str(np.random.randint(1, 99)),
                        "salary": np.random.randint(100, 15000),
                    }
                    json.dump(user_data, opened_2, indent=4)
                    opened_2.write(',')

                    index_code_1 = np.random.randint(0, len(TYPES_CODE))
                    index_code_2 = np.random.randint(0, len(TYPES_CODE))
                    index_code_3 = np.random.randint(0, len(TYPES_CODE))
                    repairs_data = {
                        "_id": i + 30,
                        "rdate": str(np.random.randint(1, 30)) + "." + str(np.random.randint(1, 12)) + "." +
                        str(np.random.randint(1, 99)),
                        "auto_num_id": i,
                        "mechanic_id": i + 20,
                        "types_code": [TYPES_CODE[index_code_1], TYPES_CODE[index_code_2],
TYPES_CODE[index_code_3]],
                    }
                    json.dump(repairs_data, opened_3, indent=4)
                    opened_3.write(',')
```

— TASK ONE
db.trailers_park_data.insertMany([<insert data from generated trailers_park_data json file >])
db.stuff_data.insertMany([<insert data from generated stuff_data json file >])
db.repairs_data.insertMany([<insert data from generated repairs_data json file >])

-- TASK TWO
-- count documents
```
db.stuff_data.aggregate([
  {
   $group : {
     _id: null,
     count: {$sum: 1}
   }
  },
])
```



-- group by surname and count ids (push them to the list)
```
db.stuff_data.aggregate([
  {
   $group : {
    _id: "$surname",
    ids: {$push: "$_id"}
   }
  },
  {
   $sort : {
     _id: -1
   }
  }
])
```



```
-- join to trailer_park_data by id
db.repairs_data.aggregate([
  {
   $lookup:
     {
      from: "trailers_park_data",
      localField: "auto_num_id",
      foreignField: "_id",
      as: "auto_num_id"
     }
  },
  {
   $project:
     {
      "_id": "$_id",
      "repair_date": "$rdate",
      "auto_brand": "$auto_num_id.auto_brand",
      "auto_num": "$auto_num_id.auto_num",
      "repairs": "$types_code"
     }
  },
  {$unwind: "$auto_brand"},
  {$unwind: "$auto_num"},
  {
   $sort: { "auto_brand": 1 }
  }
])
```

```
> db.repairs_data.aggregate([
...   {
...     $lookup:
...       {
...         from: "trailers_park_data",
...         localField: "auto_num_id",
...         foreignField: "_id",
...         as: "auto_num_id"
...       }
...   },
...   {
...     $project:
...       {
...         "_id": "$_id",
...         "repair_date": "$rdate",
...         "auto_brand": "$auto_num_id.auto_brand",
...         "auto_num": "$auto_num_id.auto_num",
...         "repairs": "$types_code"
...       }
...   },
...   {$unwind: "$auto_brand"},
...   {$unwind: "$auto_num"},
...   {
...     $sort: { "auto_brand": 1 }
...   }
... ])
{ "_id" : 30, "repair_date" : "19.4.61", "auto_brand" : "BMW", "auto_num" : 6426, "repairs" : [ "Paint wheels", "Large fixes", "Improve handle" ] }
{ "_id" : 36, "repair_date" : "7.3.13", "auto_brand" : "BMW", "auto_num" : 6084, "repairs" : [ "Full cleaning", "Large fixes", "Inside cleaning" ] }
{ "_id" : 35, "repair_date" : "25.6.16", "auto_brand" : "Ferrari", "auto_num" : 1149, "repairs" : [ "Small fixes", "Full cleaning", "Paint base" ] }
{ "_id" : 34, "repair_date" : "17.2.13", "auto_brand" : "Ford", "auto_num" : 6812, "repairs" : [ "Middle fixes", "Paint wheels", "Recolor" ] }
{ "_id" : 31, "repair_date" : "20.1.56", "auto_brand" : "Mercedes", "auto_num" : 9806, "repairs" : [ "Outside cleaning", "Repair wheels", "Inside cleaning" ] }
{ "_id" : 38, "repair_date" : "9.1.51", "auto_brand" : "Mercedes", "auto_num" : 5322, "repairs" : [ "Full cleaning", "Outside cleaning", "Middle fixes" ] }
{ "_id" : 33, "repair_date" : "9.11.12", "auto_brand" : "Schkoda", "auto_num" : 6362, "repairs" : [ "Large fixes", "Improve handle", "Full cleaning" ] }
{ "_id" : 39, "repair_date" : "15.5.9", "auto_brand" : "Subaru", "auto_num" : 1334, "repairs" : [ "Outside cleaning", "Outside cleaning", "Outside cleaning" ] }
{ "_id" : 37, "repair_date" : "9.2.88", "auto_brand" : "Suzuki", "auto_num" : 7293, "repairs" : [ "Repair wheels", "Paint base", "Large fixes" ] }
{ "_id" : 32, "repair_date" : "12.3.50", "auto_brand" : "Volkswagen", "auto_num" : 6533, "repairs" : [ "Middle fixes", "Recolor", "Small fixes" ] }
>
```

```
-- output mechanic and car. Print out also if car was cleaned. Count total salary and split on categories
db.repairs_data.aggregate([
  {
   $lookup: {
     from: "trailers_park_data",
     localField: "auto_num_id",
     foreignField: "_id",
     as: "auto_num_id"
   }
  },
  {
   $lookup: {
     from: "stuff_data",
     localField: "mechanic_id",
     foreignField: "_id",
     as: "mechanic_id"
   }
  },
  {
   $project: {
     "mechanic": "$mechanic_id.surname",
     "salary": "$mechanic_id.salary",
     "car": "$auto_num_id.auto_brand",
     "was_fully_cleaned": {
       $in : ["Full cleaning", "$types_code"]
     }
   }
  },
  {$unwind: "$mechanic"},
  {$unwind: "$car"},
  {$unwind: "$salary"},
  {
   $bucket:  {
     groupBy: "$salary",
     boundaries: [0, 5000, 10000, 20000],
     default: "Other",
     output: {
       "count": { $sum: 1 },
       "mechanics": { $push: "$mechanic" },
       "cars": { $push: "$car" },
       "total_salary": { $sum: "$salary"}
     }
   }
  }
])
```

```
> db.repairs_data.aggregate([
...   {
...     $lookup: {
...       from: "trailers_park_data",
...       localField: "auto_num_id",
...       foreignField: "_id",
...       as: "auto_num_id"
...     }
...   },
...   {
...     $lookup: {
...       from: "stuff_data",
...       localField: "mechanic_id",
...       foreignField: "_id",
...       as: "mechanic_id"
...     }
...   },
...   {
...     $project: {
...       "mechanic": "$mechanic_id.surname",
...       "salary": "$mechanic_id.salary",
...       "car": "$auto_num_id.auto_brand",
...       "was_fully_cleaned": {
...         $in : ["Full cleaning", "$types_code"]
...       }
...     }
...   },
...   {$unwind: "$mechanic"},
...   {$unwind: "$car"},
...   {$unwind: "$salary"},
...   {
...     $bucket:  {
...       groupBy: "$salary",
...       boundaries: [0, 5000, 10000, 20000],
...       default: "Other",
...       output: {
...         "count": { $sum: 1 },
...         "mechanics": { $push: "$mechanic" },
...         "cars": { $push: "$car" },
...         "total_salary": { $sum: "$salary"}
...       }
...     }
...   }
... ])
{ "_id" : 0, "count" : 2, "mechanics" : [ "Mitchel", "Frank" ], "cars" : [ "Ford", "Suzuki" ], "total_salary" : 7853 }
{ "_id" : 5000, "count" : 4, "mechanics" : [ "Mark", "Mitchel", "Brad", "Louise" ], "cars" : [ "BMW", "Mercedes", "Mercedes", "Subaru" ], "total_salary" : 30929 }
{ "_id" : 10000, "count" : 4, "mechanics" : [ "Nick", "Sofie", "Bruce", "Clark" ], "cars" : [ "Volkswagen", "Schkoda", "Ferrari", "BMW" ], "total_salary" : 47615 }
```

-- TASK THREE
----total salary per mechanic

```
var mapFunction = function() {
   emit(this._id, this.salary);
};

var reduceFunction = function(id, salary) {
   return Array.sum(salary);
};

db.stuff_data.mapReduce(
                mapFunction,
                reduceFunction,
                { out: "map_reduce_result" }
              )
```

```
> var mapFunction = function() {
...     emit(this._id, this.salary);
... };
>
> var reduceFunction = function(id, salary) {
...     return Array.sum(salary);
... };
>
> db.stuff_data.mapReduce(
...                     mapFunction,
...                     reduceFunction,
...                     { out: "map_reduce_result" }
...                   )
{
        "result" : "map_reduce_result",
        "timeMillis" : 48,
        "counts" : {
                "input" : 10,
                "emit" : 10,
                "reduce" : 0,
                "output" : 10
        },
        "ok" : 1
}
>
> db.map_reduce_result.find()
{ "_id" : 20, "value" : 5397 }
{ "_id" : 21, "value" : 8772 }
{ "_id" : 22, "value" : 11838 }
{ "_id" : 23, "value" : 12440 }
{ "_id" : 24, "value" : 3602 }
{ "_id" : 25, "value" : 10682 }
{ "_id" : 26, "value" : 12655 }
{ "_id" : 27, "value" : 4251 }
{ "_id" : 28, "value" : 9147 }
{ "_id" : 29, "value" : 7613 }
>
```

```
-- one more query
var mapFoo = function() {
  emit(this._id, this.released_date);
};

var reduceFoo = function(id, date){
  date_splitted = date.split('.');
  var day, month, year;
  day = date_splitted[0];
  month = date_splitted[1];
  year = date_splitted[2];

  return (year > 50) ? year : null
};


db.trailers_park_data.mapReduce(
                mapFoo,
                reduceFoo,
                { out: "map_reduce_date" }
               )
```

```
> var mapFoo = function() {
...   emit(this._id, this.released_date);
... };
>
> var reduceFoo = function(id, date){
...   date_splitted = date.split('.');
...   var day, month, year;
...   day = date_splitted[0];
...   month = date_splitted[1];
...   year = date_splitted[2];
...
...   return (year > 50) ? year : null
... };
>
>
> db.trailers_park_data.mapReduce(
...                     mapFoo,
...                     reduceFoo,
...                     { out: "map_reduce_date" }
...                   )
{
        "result" : "map_reduce_date",
        "timeMillis" : 52,
        "counts" : {
                "input" : 10,
                "emit" : 10,
                "reduce" : 0,
                "output" : 10
        },
        "ok" : 1
}
> db.map_reduce_date.find()
{ "_id" : 0, "value" : "23.1.51" }
{ "_id" : 1, "value" : "14.11.17" }
{ "_id" : 2, "value" : "15.1.22" }
{ "_id" : 3, "value" : "11.1.60" }
{ "_id" : 4, "value" : "14.6.2" }
{ "_id" : 5, "value" : "4.6.72" }
{ "_id" : 6, "value" : "21.8.63" }
{ "_id" : 7, "value" : "28.11.70" }
{ "_id" : 8, "value" : "27.5.30" }
{ "_id" : 9, "value" : "4.5.73" }
>
```