Assignment No 8:   Aim: Implementation of MongoDB Aggregation.

1.Display only the city name & its population for all the cities in state “VT”

```
>db.cities.aggregate([
{  $match:{state:"VT"} },
{
 $project:
   {_id:0, city:1, pop:1 }
}
]);
    { "city" : "BROOKFIELD", "pop" : 453 }
    { "city" : "BROWNSVILLE", "pop" : 415 }
    { "city" : "CHELSEA", "pop" : 1022 }
    { "city" : "CORINTH", "pop" : 1035 }
    { "city" : "EAST CORINTH", "pop" : 279 }
    { "city" : "EAST RANDOLPH", "pop" : 322 }
    { "city" : "RYEGATE", "pop" : 328 }
```

2. Count the number of cities in each state.

```
>db.cities.aggregate([
{   $group:
  {_id:{state:"$state"},
     noOfCities:{$sum:1}}
}
])
{ "_id" : { "state" : "RI" }, "noOfCities" : 5 }
{ "_id" : { "state" : "NY" }, "noOfCities" : 10 }
{ "_id" : { "state" : "CT" }, "noOfCities" : 8 }
{ "_id" : { "state" : "VT" }, "noOfCities" : 7 }
{ "_id" : { "state" : "NH" }, "noOfCities" : 6 }
{ "_id" : { "state" : "NJ" }, "noOfCities" : 8 }
{ "_id" : { "state" : "ME" }, "noOfCities" : 7 }
{ "_id" : { "state" : "MA" }, "noOfCities" : 5 }
```

3. Display the names of top 10 cities with largest population along with state name.

```
> db.cities.aggregate([
{ $sort: { pop: -1 } },
{ $limit: 10 },
{ $project: { _id:0, state:1, city:1, pop:1 }}
 ] )
{ "city" : "NEW YORK", "pop" : 84143, "state" : "NY" }
{ "city" : "BAYONNE", "pop" : 61444, "state" : "NJ" }
{ "city" : "BRISTOL", "pop" : 60670, "state" : "CT" }
{ "city" : "NEW YORK", "pop" : 57426, "state" : "NY" }
{ "city" : "NEW YORK", "pop" : 51224, "state" : "NY" }
{ "city" : "NEW YORK", "pop" : 46560, "state" : "NY" }
{ "city" : "BLOOMFIELD", "pop" : 46131, "state" : "NJ" }
{ "city" : "CUSHMAN", "pop" : 36963, "state" : "MA" }
{ "city" : "WEST CALDWELL", "pop" : 24946, "state" : "NJ" }
{ "city" : "NEW YORK", "pop" : 24907, "state" : "NY" }
>
```

4. Display the name of state and its average population

```
> db.cities.aggregate([
   {  $group:
      {_id:"$state", avg_pop:{  $avg:"$pop"} }
}
]);
{ "_id" : "MA", "avg_pop" : 13733.2 }
{ "_id" : "RI", "avg_pop" : -1371.8 }
{ "_id" : "CT", "avg_pop" : 14191.375 }
{ "_id" : "NJ", "avg_pop" : 24976.75 }
{ "_id" : "NY", "avg_pop" : 29046.1 }
{ "_id" : "VT", "avg_pop" : 550.5714285714286 }
{ "_id" : "ME", "avg_pop" : 4120.571428571428 }
{ "_id" : "NH", "avg_pop" : 9144.166666666666 }
```

5. Display city name and population of first 3 cities of state of MH with highest population

```
> db.cities.aggregate([
 { $match: {state:"NH"} },
 { $sort: { pop:-1} },
{ $limit: 3 },
{ $project:{ _id:0, pop:1, city:1  }} ] )
{ "city" : "AMHERST", "pop" : 13998 }
{ "city" : "AUBURN", "pop" : 9085 }
{ "city" : "CANDIA", "pop" : 8557 }
>
```

6. Display list of all cities in “CT” state in descending order of population

```
> db.cities.aggregate([
... {  $match:  { state:"CT" } },
... {  $sort: { pop: -1 } },
... {  $project: { _id:0, pop:1, city:1 }}
... ])
```

```
{ "city" : "BRISTOL", "pop" : 60670, "state" : "CT" }
{ "city" : "BLOOMFIELD", "pop" : 19524, "state" : "CT" }
{ "city" : "AVON", "pop" : 13988, "state" : "CT" }
{ "city" : "BURLINGTON", "pop" : 7017, "state" : "CT" }
{ "city" : "WINDSORVILLE", "pop" : 5067, "state" : "CT" }
{ "city" : "CANTON", "pop" : 4125, "state" : "CT" }
{ "city" : "CANAAN", "pop" : 2948, "state" : "CT" }
{ "city" : "CANTON CENTER", "pop" : 192, "state" : "CT" }
>
```

7. Map-reduce function to find the different keys(i.e. fields like city name, pop, state etc) in the collection and count its occurrence

```
> var mapper1 = function() { for( key in this ) { emit(key,1); } };
> var reduce1 = function(field,count) { return Array.sum(count); };
> db.cities.mapReduce(mapper1,reduce1, {out : "query_fields"})
{ "result" : "query_fields", "ok" : 1 }
> db.query_fields.find()
{ "_id" : "city", "value" : 56 }
{ "_id" : "loc", "value" : 56 }
{ "_id" : "_id", "value" : 56 }
{ "_id" : "state", "value" : 56 }
{ "_id" : "pop", "value" : 56 }
```