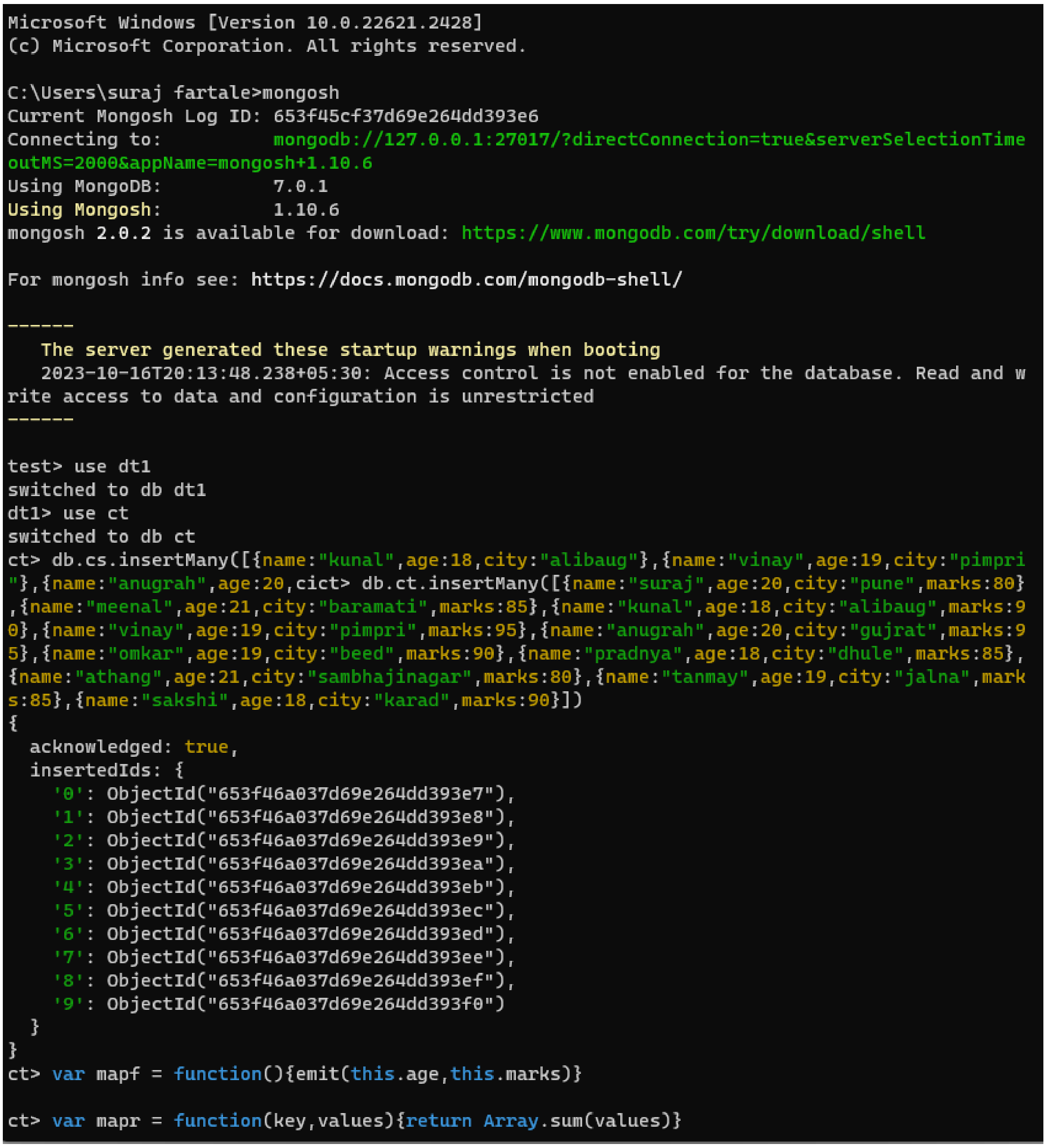
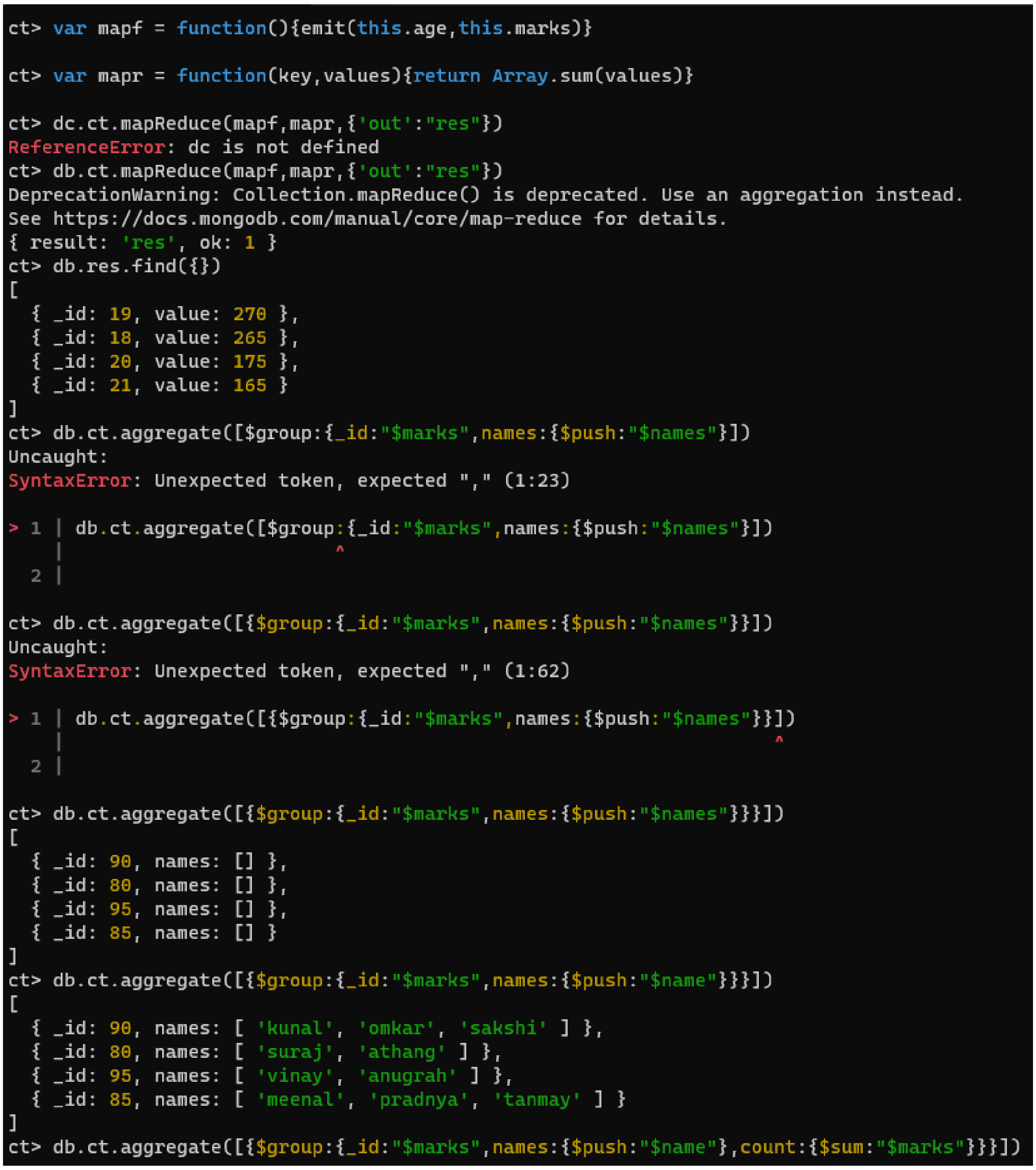
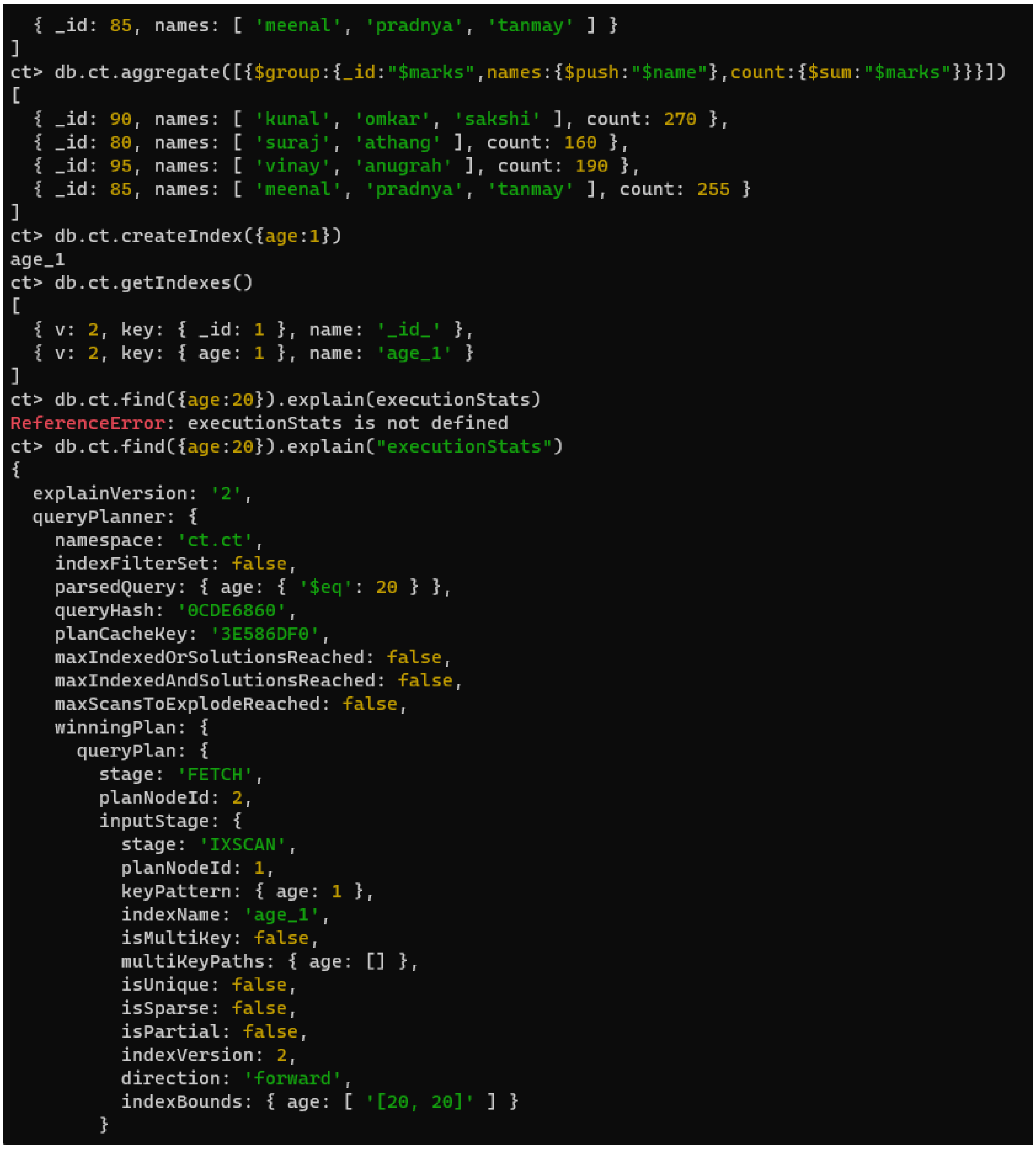


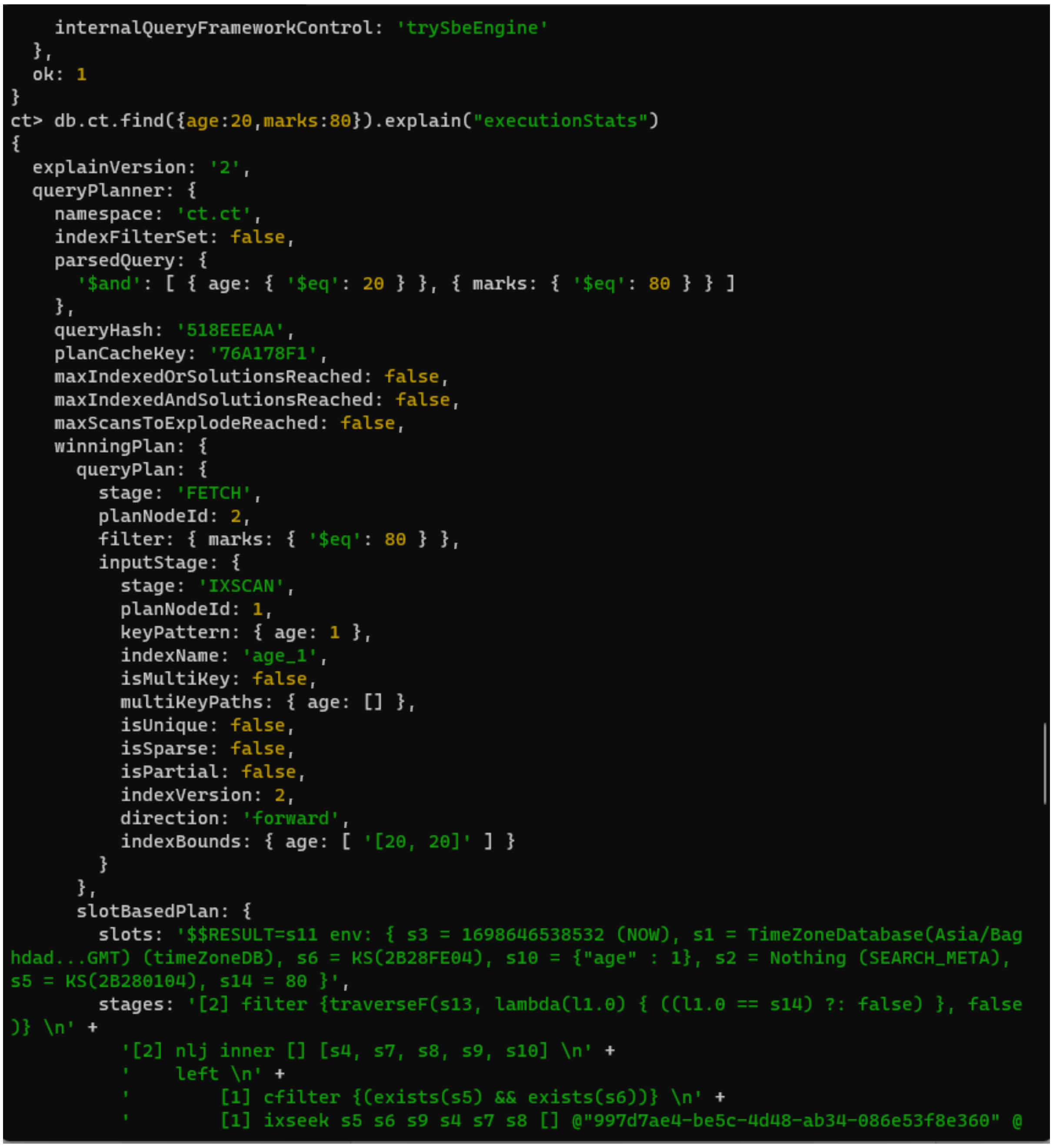
```
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\suraj fartale>mongosh
Current Mongosh Log ID: 653f45cf37d69e264dd393e6
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTime
outMS=2000&appName=mongosh+1.10.6
Using MongoDB:          7.0.1
Using Mongosh:          1.10.6
mongosh 2.0.2 is available for download: https://www.mongodb.com/try/download/shell

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

------
   The server generated these startup warnings when booting
   2023-10-16T20:13:48.238+05:30: Access control is not enabled for the database. Read and w
rite access to data and configuration is unrestricted
------

test> use dt1
switched to db dt1
dt1> use ct
switched to db ct
ct> db.cs.insertMany([{name:"kunal",age:18,city:"alibaug"},{name:"vinay",age:19,city:"pimpri
"},{name:"anugrah",age:20,cict> db.ct.insertMany([{name:"suraj",age:20,city:"pune",marks:80}
,{name:"meenal",age:21,city:"baramati",marks:85},{name:"kunal",age:18,city:"alibaug",marks:9
0},{name:"vinay",age:19,city:"pimpri",marks:95},{name:"anugrah",age:20,city:"gujrat",marks:9
5},{name:"omkar",age:19,city:"beed",marks:90},{name:"pradnya",age:18,city:"dhule",marks:85},
{name:"athang",age:21,city:"sambhajinagar",marks:80},{name:"tanmay",age:19,city:"jalna",mark
s:85},{name:"sakshi",age:18,city:"karad",marks:90}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("653f46a037d69e264dd393e7"),
    '1': ObjectId("653f46a037d69e264dd393e8"),
    '2': ObjectId("653f46a037d69e264dd393e9"),
    '3': ObjectId("653f46a037d69e264dd393ea"),
    '4': ObjectId("653f46a037d69e264dd393eb"),
    '5': ObjectId("653f46a037d69e264dd393ec"),
    '6': ObjectId("653f46a037d69e264dd393ed"),
    '7': ObjectId("653f46a037d69e264dd393ee"),
    '8': ObjectId("653f46a037d69e264dd393ef"),
    '9': ObjectId("653f46a037d69e264dd393f0")
  }
}
ct> var mapf = function(){emit(this.age,this.marks)}

ct> var mapr = function(key,values){return Array.sum(values)}
```



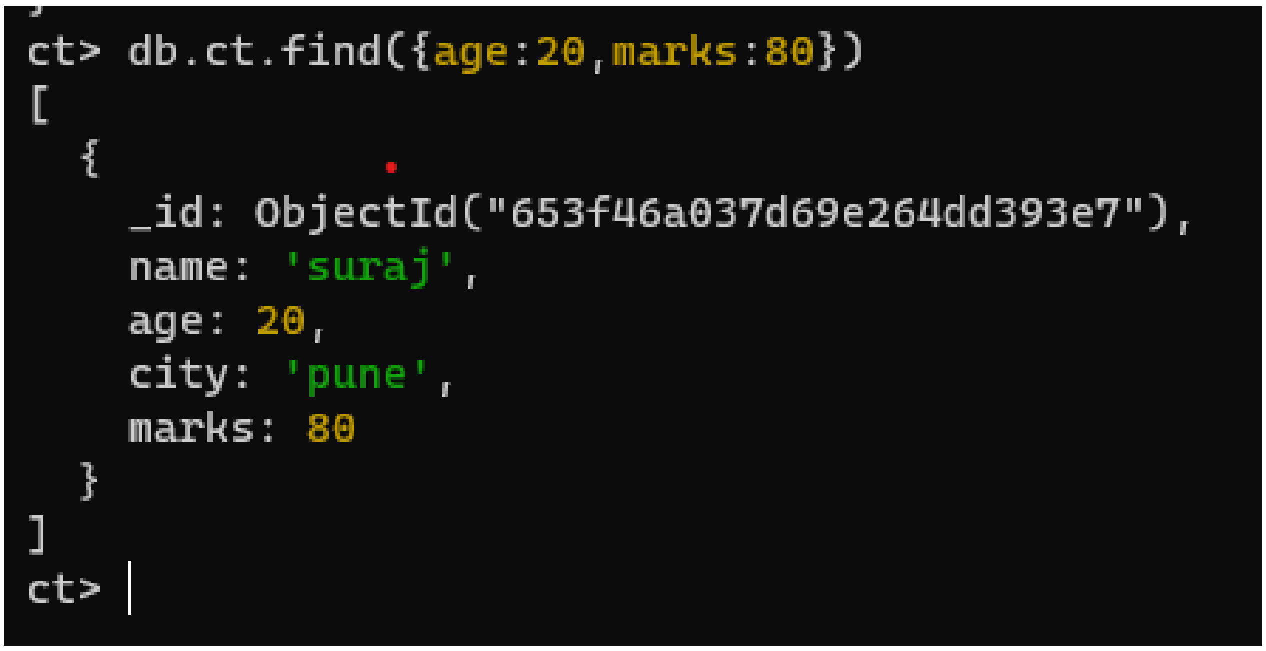
```
ct> var mapf = function(){emit(this.age,this.marks)}

ct> var mapr = function(key,values){return Array.sum(values)}

ct> dc.ct.mapReduce(mapf,mapr,{'out':"res"})
ReferenceError: dc is not defined
ct> db.ct.mapReduce(mapf,mapr,{'out':"res"})
DeprecationWarning: Collection.mapReduce() is deprecated. Use an aggregation instead.
See https://docs.mongodb.com/manual/core/map-reduce for details.
{ result: 'res', ok: 1 }
ct> db.res.find({})
[
  { _id: 19, value: 270 },
  { _id: 18, value: 265 },
  { _id: 20, value: 175 },
  { _id: 21, value: 165 }
]
ct> db.ct.aggregate([$group:{_id:"$marks",names:{$push:"$names"}])
Uncaught:
SyntaxError: Unexpected token, expected "," (1:23)

> 1 | db.ct.aggregate([$group:{_id:"$marks",names:{$push:"$names"}])
    |                        ^
  2 |

ct> db.ct.aggregate([{$group:{_id:"$marks",names:{$push:"$names"}}])
Uncaught:
SyntaxError: Unexpected token, expected "," (1:62)

> 1 | db.ct.aggregate([{$group:{_id:"$marks",names:{$push:"$names"}}])
    |                                                               ^
  2 |

ct> db.ct.aggregate([{$group:{_id:"$marks",names:{$push:"$names"}}}])
[
  { _id: 90, names: [] },
  { _id: 80, names: [] },
  { _id: 95, names: [] },
  { _id: 85, names: [] }
]
ct> db.ct.aggregate([{$group:{_id:"$marks",names:{$push:"$name"}}}])
[
  { _id: 90, names: [ 'kunal', 'omkar', 'sakshi' ] },
  { _id: 80, names: [ 'suraj', 'athang' ] },
  { _id: 95, names: [ 'vinay', 'anugrah' ] },
  { _id: 85, names: [ 'meenal', 'pradnya', 'tanmay' ] }
]
ct> db.ct.aggregate([{$group:{_id:"$marks",names:{$push:"$name"},count:{$sum:"$marks"}}}])
```



```
  { _id: 85, names: [ 'meenal', 'pradnya', 'tanmay' ] }
]
ct> db.ct.aggregate([{$group:{_id:"$marks",names:{$push:"$name"},count:{$sum:"$marks"}}}])
[
  { _id: 90, names: [ 'kunal', 'omkar', 'sakshi' ], count: 270 },
  { _id: 80, names: [ 'suraj', 'athang' ], count: 160 },
  { _id: 95, names: [ 'vinay', 'anugrah' ], count: 190 },
  { _id: 85, names: [ 'meenal', 'pradnya', 'tanmay' ], count: 255 }
]
ct> db.ct.createIndex({age:1})
age_1
ct> db.ct.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { age: 1 }, name: 'age_1' }
]
ct> db.ct.find({age:20}).explain(executionStats)
ReferenceError: executionStats is not defined
ct> db.ct.find({age:20}).explain("executionStats")
{
  explainVersion: '2',
  queryPlanner: {
    namespace: 'ct.ct',
    indexFilterSet: false,
    parsedQuery: { age: { '$eq': 20 } },
    queryHash: '0CDE6860',
    planCacheKey: '3E586DF0',
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      queryPlan: {
        stage: 'FETCH',
        planNodeId: 2,
        inputStage: {
          stage: 'IXSCAN',
          planNodeId: 1,
          keyPattern: { age: 1 },
          indexName: 'age_1',
          isMultiKey: false,
          multiKeyPaths: { age: [] },
          isUnique: false,
          isSparse: false,
          isPartial: false,
          indexVersion: 2,
          direction: 'forward',
          indexBounds: { age: [ '[20, 20]' ] }
        }
```



```
    internalQueryFrameworkControl: 'trySbeEngine'
  },
  ok: 1
}
ct> db.ct.find({age:20,marks:80}).explain("executionStats")
{
  explainVersion: '2',
  queryPlanner: {
    namespace: 'ct.ct',
    indexFilterSet: false,
    parsedQuery: {
      '$and': [ { age: { '$eq': 20 } }, { marks: { '$eq': 80 } } ]
    },
    queryHash: '518EEEAA',
    planCacheKey: '76A178F1',
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      queryPlan: {
        stage: 'FETCH',
        planNodeId: 2,
        filter: { marks: { '$eq': 80 } },
        inputStage: {
          stage: 'IXSCAN',
          planNodeId: 1,
          keyPattern: { age: 1 },
          indexName: 'age_1',
          isMultiKey: false,
          multiKeyPaths: { age: [] },
          isUnique: false,
          isSparse: false,
          isPartial: false,
          indexVersion: 2,
          direction: 'forward',
          indexBounds: { age: [ '[20, 20]' ] }
        }
      },
      slotBasedPlan: {
        slots: '$$RESULT=s11 env: { s3 = 1698646538532 (NOW), s1 = TimeZoneDatabase(Asia/Bag
hdad...GMT) (timeZoneDB), s6 = KS(2B28FE04), s10 = {"age" : 1}, s2 = Nothing (SEARCH_META),
s5 = KS(2B280104), s14 = 80 }',
        stages: '[2] filter {traverseF(s13, lambda(l1.0) { ((l1.0 == s14) ?: false) }, false
)} \n' +
          '[2] nlj inner [] [s4, s7, s8, s9, s10] \n' +
          '    left \n' +
          '        [1] cfilter {(exists(s5) && exists(s6))} \n' +
          '        [1] ixseek s5 s6 s9 s4 s7 s8 [] @"997d7ae4-be5c-4d48-ab34-086e53f8e360" @
```



```
ct> db.ct.find({age:20,marks:80})
[
  {
    _id: ObjectId("653f46a037d69e264dd393e7"),
    name: 'suraj',
    age: 20,
    city: 'pune',
    marks: 80
  }
]
ct>
```