

```

> db.createCollection("student")
{ "ok" : 1 }
> db.student.insert({"roll":10, "name":"clavin", "contact":123})
WriteResult({ "nInserted" : 1 })

> db.student.find()
{ "_id" : ObjectId("57b40b36d6a4814fd5292e63"), "roll" : 10, "name" : "clavin", "contact" : 123, "address" : "Pune" }
{ "_id" : ObjectId("57b40bc8d6a4814fd5292e64"), "roll" : 11, "name" : "justin", "contact" : 456, "address" : "Pune" }
{ "_id" : ObjectId("57b40bc8d6a4814fd5292e65"), "roll" : 14, "name" : "michel", "contact" : 888, "address" : "Mumbai" }
{ "_id" : ObjectId("57b40bc8d6a4814fd5292e66"), "roll" : 15, "name" : "roger", "contact" : 789, "address" : "Delhi" }
{ "_id" : ObjectId("57b40bc8d6a4814fd5292e67"), "roll" : 17, "name" : "usain", "contact" : 958, "address" : "Delhi" }
{ "_id" : ObjectId("57b40bc8d6a4814fd5292e69"), "roll" : 22, "name" : "peter", "contact" : 999, "address" : "Pune" }
{ "_id" : ObjectId("57b40bc8d6a4814fd5292e6a"), "roll" : 23, "name" : "james", "contact" : 878, "address" : "Kolkata" }
{ "_id" : ObjectId("57b40ddb6a4814fd5292e6b"), "roll" : 25, "name" : { "fname" : "bill", "lname" : "clinton" }, "contact" : 1111, "address" : "Mumbai" }
>

```

SIMPLE INDEX:

```

> db.student.find({"name":"peter"})
{ "_id" : ObjectId("57b40bc8d6a4814fd5292e69"), "roll" : 22, "name" : "peter", "contact" : 999, "address" : "Pune" }
> db.student.find({"name":"peter"}).explain()
{
  "cursor" : "BasicCursor",
  "isMultiKey" : false,
  "n" : 1,
  "nscannedObjects" : 8,
  "nscanned" : 8,
  "nscannedObjectsAllPlans" : 8,
  "nscannedAllPlans" : 8,
  "scanAndOrder" : false,
  "indexOnly" : false,
  "nYields" : 0,
  "nChunkSkips" : 0,
  "millis" : 0,
  "server" : "localhost.localdomain:27017",
  "filterSet" : false
}
> db.student.ensureIndex({ "name" : 1 })
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
> db.student.find({"name":"peter"}).explain()
{
  "cursor" : "BtreeCursor name_1",

```

```

    "isMultiKey" : false,
    "n" : 1,
    "nscannedObjects" : 1,
    "nscanned" : 1,
    "nscannedObjectsAllPlans" : 1,
    "nscannedAllPlans" : 1,
    "scanAndOrder" : false,
    "indexOnly" : false,
    "nYields" : 0,
    "nChunkSkips" : 0,
    "millis" : 0,
    "indexBounds" : {
      "name" : [
        [
          "peter",
          "peter"
        ]
      ]
    },
    "server" : "localhost.localdomain:27017",
    "filterSet" : false
  }
}
>

```

UNIQUE: Act as primary key ,avoids duplication of document

```

> db.student.find({"roll":17})
> db.student.find({"roll":17})
{ "_id" : ObjectId("57b40bc8d6a4814fd5292e67"), "roll" : 17, "name" : "usain", "contact" : 958, "address" :
"Delhi" }
> db.student.find({"roll":17}).explain()
{
  "cursor" : "BasicCursor",
  "isMultiKey" : false,
  "n" : 1,
  "nscannedObjects" : 8,
  "nscanned" : 8,
  "nscannedObjectsAllPlans" : 8,
  "nscannedAllPlans" : 8,
  "scanAndOrder" : false,
  "indexOnly" : false,
  "nYields" : 0,
  "nChunkSkips" : 0,
  "millis" : 0,
  "server" : "localhost.localdomain:27017",
  "filterSet" : false
}
> db.student.ensureIndex( { "roll": 1 }, { unique: true } )
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 2,
  "numIndexesAfter" : 3,
  "ok" : 1
}
> db.student.find({"roll":17}).explain()
{

```

```

        "cursor" : "BtreeCursor roll_1",
        "isMultiKey" : false,
        "n" : 1,
        "nscannedObjects" : 1,
        "nscanned" : 1,
        "nscannedObjectsAllPlans" : 1,
        "nscannedAllPlans" : 1,
        "scanAndOrder" : false,
        "indexOnly" : false,
        "nYields" : 0,
        "nChunkSkips" : 0,
        "millis" : 0,
        "indexBounds" : {
            "roll" : [
                [
                    17,
                    17
                ]
            ]
        },
        "server" : "localhost.localdomain:27017",
        "filterSet" : false
    }
}
> db.student.insert({"roll":10, "name":"joker", "contact":808})
WriteResult({
  "nInserted" : 0,
  "writeError" : {
    "code" : 11000,
    "errmsg" : "insertDocument :: caused by :: 11000 E11000 duplicate key error index:
te3174db.student.$roll_1 dup key: { : 10.0 }"
  }
})
>

```

COMPOUND index:

```

> db.student.find( { $or: [ { contact: 958 }, { address:"Pune" } ] } )
{ "_id" : ObjectId("57b40b36d6a4814fd5292e63"), "roll" : 10, "name" : "clavin", "contact" : 123, "address" :
"Pune" }
{ "_id" : ObjectId("57b40bc8d6a4814fd5292e64"), "roll" : 11, "name" : "justin", "contact" : 456, "address" :
"Pune" }
{ "_id" : ObjectId("57b40bc8d6a4814fd5292e67"), "roll" : 17, "name" : "usain", "contact" : 958, "address" :
"Delhi" }
{ "_id" : ObjectId("57b40bc8d6a4814fd5292e69"), "roll" : 22, "name" : "peter", "contact" : 999, "address" :
"Pune" }
> db.student.find( { $or: [ { contact: 958 }, { address:"Pune" } ] } ).explain()
{
  "cursor" : "BasicCursor",
  "isMultiKey" : false,
  "n" : 4,
  "nscannedObjects" : 8,
  "nscanned" : 8,
  "nscannedObjectsAllPlans" : 8,
  "nscannedAllPlans" : 8,
  "scanAndOrder" : false,
  "indexOnly" : false,
  "nYields" : 0,

```

```

        "nChunkSkips" : 0,
        "millis" : 0,
        "server" : "localhost.localdomain:27017",
        "filterSet" : false
    }
}
> db.student.ensureIndex( { "contact": 1, "address": -1 } )
{
    "createdCollectionAutomatically" : false,
    "numIndexesBefore" : 3,
    "numIndexesAfter" : 4,
    "ok" : 1
}
> db.student.find( { $or: [ { contact: 958 }, { address:"Pune" } ] } )
{ "_id" : ObjectId("57b40b36d6a4814fd5292e63"), "roll" : 10, "name" : "clavin", "contact" : 123, "address" :
"Pune" }
{ "_id" : ObjectId("57b40bc8d6a4814fd5292e64"), "roll" : 11, "name" : "justin", "contact" : 456, "address" :
"Pune" }
{ "_id" : ObjectId("57b40bc8d6a4814fd5292e67"), "roll" : 17, "name" : "usain", "contact" : 958, "address" :
"Delhi" }
{ "_id" : ObjectId("57b40bc8d6a4814fd5292e69"), "roll" : 22, "name" : "peter", "contact" : 999, "address" :
"Pune" }
> db.student.find( { $or: [ { contact: 958 }, { address:"Pune" } ] } ).explain()
{
    "cursor" : "BasicCursor",
    "isMultiKey" : false,
    "n" : 4,
    "nscannedObjects" : 8,
    "nscanned" : 8,
    "nscannedObjectsAllPlans" : 8,
    "nscannedAllPlans" : 8,
    "scanAndOrder" : false,
    "indexOnly" : false,
    "nYields" : 0,
    "nChunkSkips" : 0,
    "millis" : 0,
    "server" : "localhost.localdomain:27017",
    "filterSet" : false
}
> db.student.find( { contact: 958 } ).explain()
{
    "cursor" : "BtreeCursor contact_1_address_-1",
    "isMultiKey" : false,
    "n" : 1,
    "nscannedObjects" : 1,
    "nscanned" : 1,
    "nscannedObjectsAllPlans" : 1,
    "nscannedAllPlans" : 1,
    "scanAndOrder" : false,
    "indexOnly" : false,
    "nYields" : 0,
    "nChunkSkips" : 0,
    "millis" : 0,
    "indexBounds" : {
        "contact" : [
            [
                958,
                958
            ]
        ]
    }
}

```

```

        ]
      ],
      "address" : [
        [
          {
            "$maxElement" : 1
          },
          {
            "$minElement" : 1
          }
        ]
      ]
    },
    "server" : "localhost.localdomain:27017",
    "filterSet" : false
  }
}

```

AVAILABLE INDEXES

db.student.getIndexes()

```

[
  {
    "v" : 1,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_",
    "ns" : "te3174db.student"
  },
  {
    "v" : 1,
    "key" : {
      "name" : 1
    },
    "name" : "name_1",
    "ns" : "te3174db.student"
  },
  {
    "v" : 1,
    "unique" : true,
    "key" : {
      "roll" : 1
    },
    "name" : "roll_1",
    "ns" : "te3174db.student"
  },
  {
    "v" : 1,
    "key" : {
      "contact" : 1,
      "address" : -1
    },
    "name" : "contact_1_address_-1",
    "ns" : "te3174db.student"
  }
]

```

Aggregation:

SUM

```
> db.student.aggregate([{$group:{_id:"$address",total:{$sum:"$roll"}}}])
{ "_id" : "Kolkata", "total" : 23 }
{ "_id" : "Delhi", "total" : 32 }
{ "_id" : "Mumbai", "total" : 39 }
{ "_id" : "Pune", "total" : 43 }
```

MAX

```
> db.student.aggregate([{$group:{_id:"$address",total:{$max:"$roll"}}}])
{ "_id" : "Kolkata", "total" : 23 }
{ "_id" : "Delhi", "total" : 17 }
{ "_id" : "Mumbai", "total" : 25 }
{ "_id" : "Pune", "total" : 22 }
```

MIN

```
> db.student.aggregate([{$group:{_id:"$address",total:{$min:"$roll"}}}])
{ "_id" : "Kolkata", "total" : 23 }
{ "_id" : "Delhi", "total" : 15 }
{ "_id" : "Mumbai", "total" : 14 }
{ "_id" : "Pune", "total" : 10 }
```

>

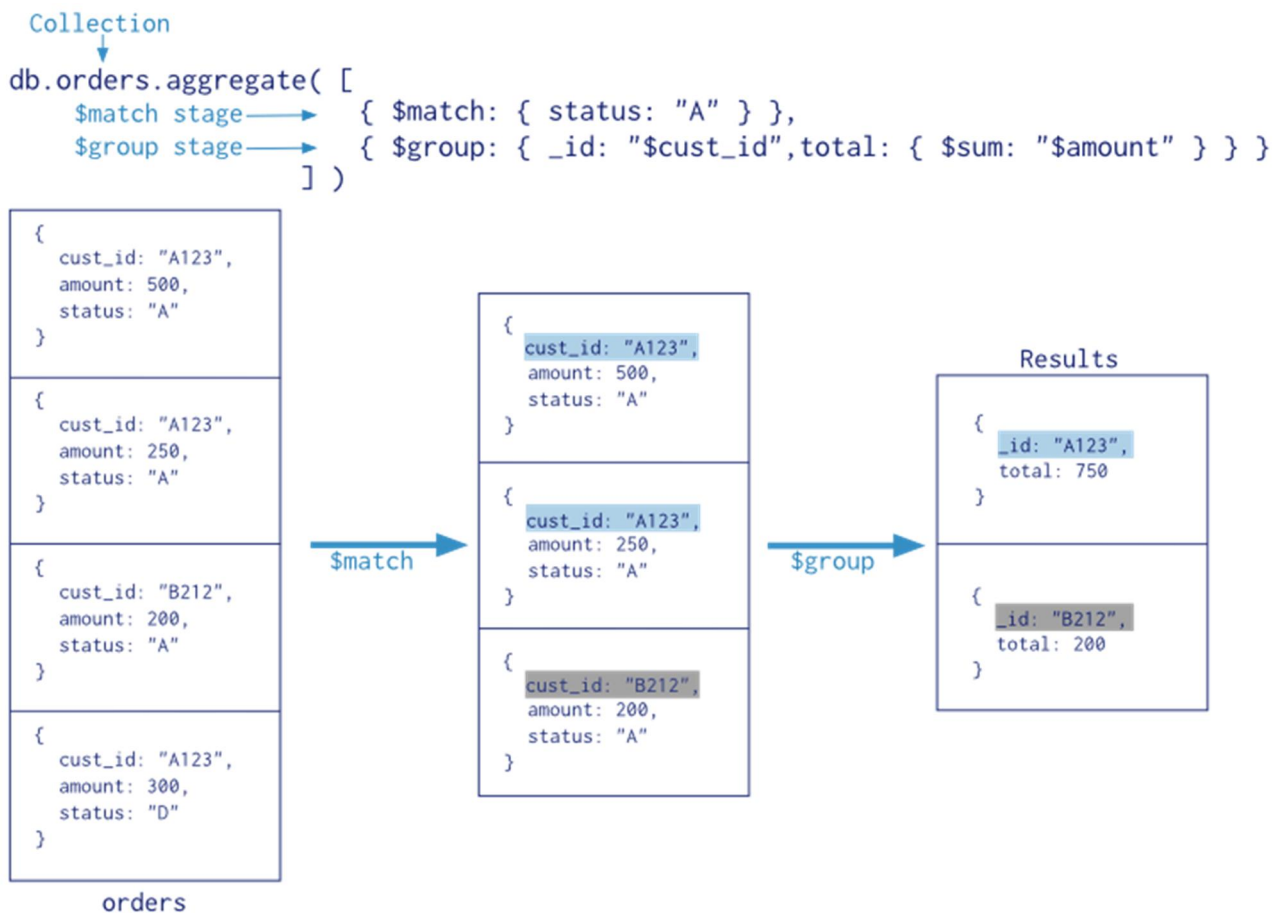
MATCH

```
> db.student.aggregate([{$match:{address:"Pune"}}])
{ "_id" : ObjectId("57b40b36d6a4814fd5292e63"), "roll" : 10, "name" : "clavin", "contact" : 123, "address" : "Pune" }
{ "_id" : ObjectId("57b40bc8d6a4814fd5292e64"), "roll" : 11, "name" : "justin", "contact" : 456, "address" : "Pune" }
{ "_id" : ObjectId("57b40bc8d6a4814fd5292e69"), "roll" : 22, "name" : "peter", "contact" : 999, "address" : "Pune" }
```

>

AGGREGATION with PIPELINE

The aggregation pipeline is a framework for data aggregation modeled on the concept of data processing pipelines. Documents enter a multi-stage pipeline that transforms the documents into aggregated results.



The aggregation pipeline provides an alternative to *map-reduce* and may be the preferred solution for aggregation tasks where the complexity of map-reduce may be unwarranted. Aggregation pipeline have some limitations on value types and result size

Pipeline

The MongoDB aggregation pipeline consists of *stages*. Each stage transforms the documents as they pass through the pipeline. Pipeline stages do not need to produce one output document for every input document; e.g., some stages may generate new documents or filter out documents. Pipeline stages can appear multiple times in the pipeline.

MongoDB provides the `db.collection.aggregate()` method in the mongo shell and the `aggregate` command for aggregation pipeline. See *Stage Operators* for the available stages.

```
> db.student.aggregate([{$group:{_id:"$address",no:{$sum:1}}}] )
{ "_id" : "Kolkata", "no" : 1 }
{ "_id" : "Delhi", "no" : 2 }
{ "_id" : "Mumbai", "no" : 2 }
{ "_id" : "Pune", "no" : 3 }
>
```

```
> db.student.aggregate([{$match:{address:"Mumbai"}},{$group:{_id:"$address",no:{$sum:1}}}] )
{ "_id" : "Mumbai", "no" : 2 }
```