MongoDB
Aggregation

Team Emertxe

Aggregation

Aggregation

Aggregations operations process data that returns the computed results.

Aggregation operations groups data from multiple documents together, and can perform a variety of operations on the grouped data to return a single result.

Aggregation

MongoDB provide 3 ways to perform aggregation :

- Aggregation pipeline
- Map-Reduce
- Single purpose aggregation operations.

Aggregation Pipeline

The MongoDB pipeline consists of stages.

Each stages transforms the document as they pass.

The aggregation framework is modeled on the concept of data processing pipeline.

Documents enter a multistage pipeline that transform the document into an aggregated result.

Aggregation Pipeline

Aggregate functions is used to group the documents in a collection.

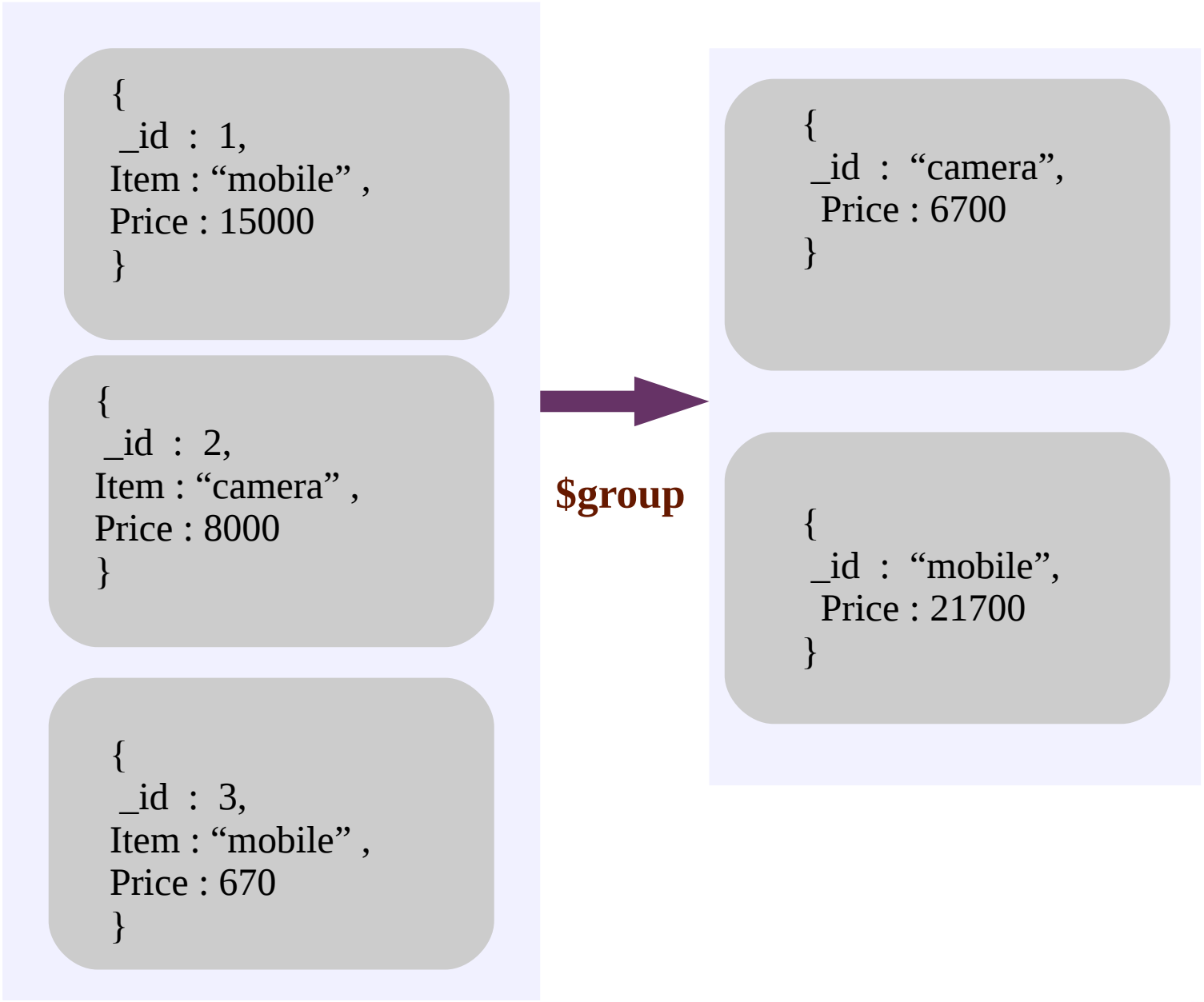
It can provide sum ,average , minimum and maximum of selected group.

In MongoDB aggregate() is used to group documents.

Syntax :

db.collection_name.aggregate(aggreagte operation)

Aggregation pipeline



EMERTXE

Aggregation Pipeline Example

```
> db.order.find({});
  { "_id" : 1, "item" : "mobile", "price" : 15000, "qty" : 2 }
  { "_id" : 2, "item" : "camera", "price" : 8000, "qty" : 3 }
  { "_id" : 3, "item" : "mobile", "price" : 6700, "qty" : 5 }
> db.order.aggregate([ { $group : { _id : "$item" , price :
    { $sum : "$price" }}}] );
  { "_id" : "camera", "price" : 8000 }
  { "_id" : "mobile", "price" : 21700 }
```

The above example will display the total price of each item.

Exercise

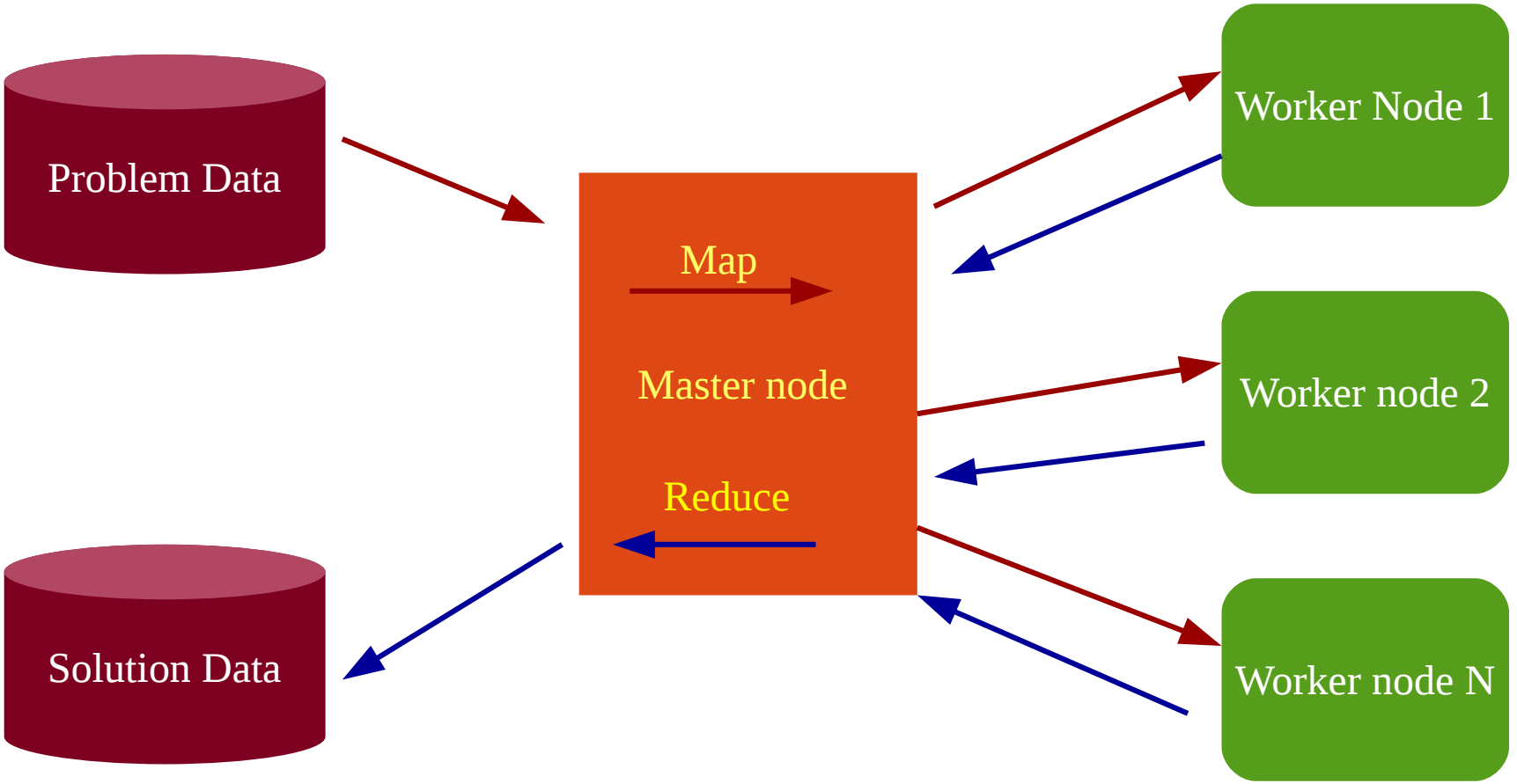
- Write a MongoDB query to display the total salary from employee collection where salary is more than 25000.
- Write a MongoDB query to display total quantity where item price is more than 8500.

Map-Reduce

Map-reduce is a data processing paradigm for condensing large volumes of data into useful aggregated results.

The mapReduce command allows to run map-reduce aggregation operations over collection.

Map-Reduce



EMERTXE

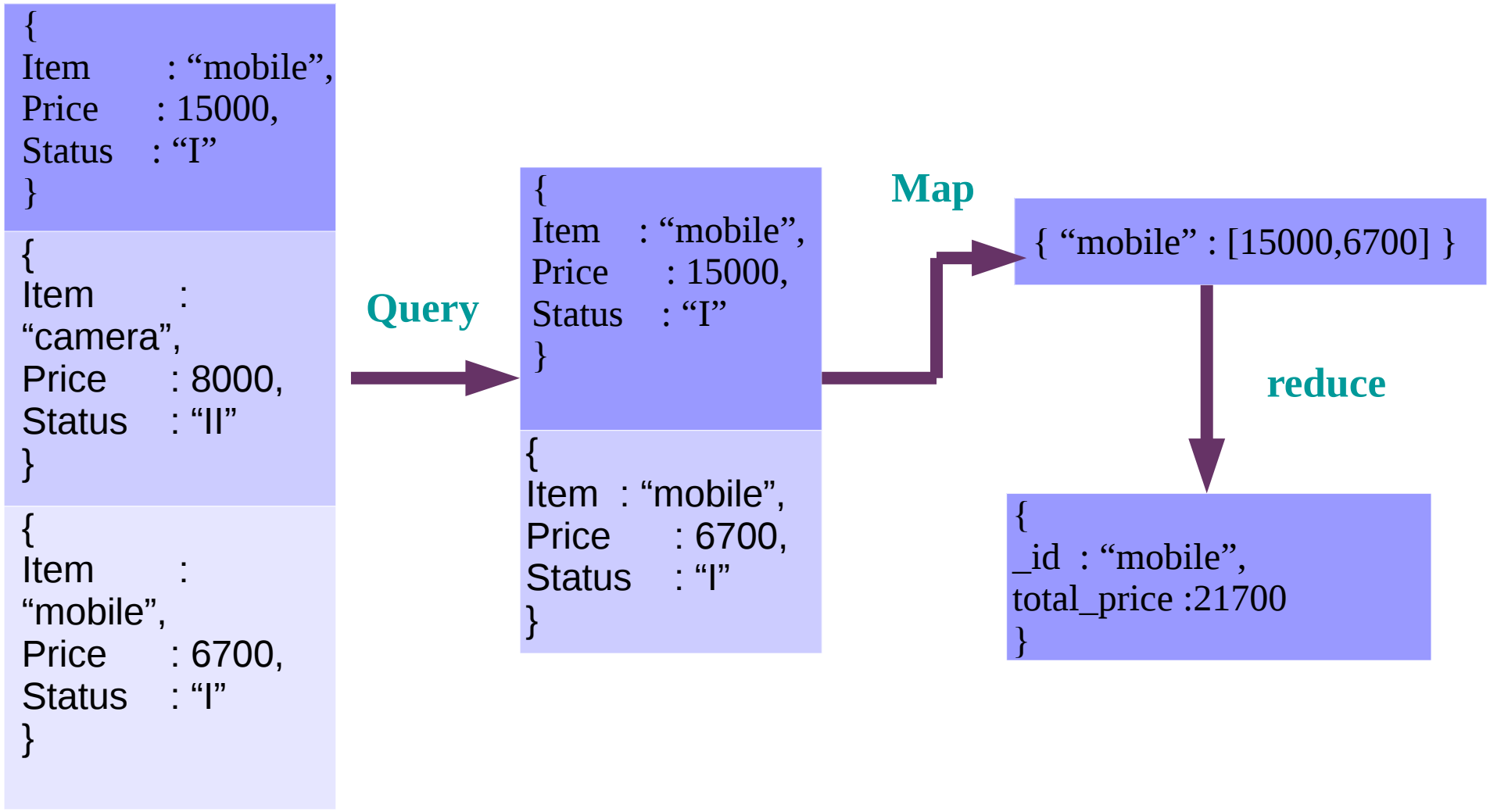
Map-Reduce operation

Map-Reduce operation takes the documents of a single collection as a input and perform query before beginning the map-phase.

The map function emits the key-value pairs.

MongoDB applies the reduce phase for those key that have multiple values.

Map-Reduce operation



Map-Reduce
Example

```
> db.order.mapReduce( function() { emit (this.id , this.price ); } ,
  function (key ,values ) { return Array.sum (values) } ,
 {query : { status : "I" } ,  out : "total_price" })
 {
  "result" : "total_price",
  "timeMillis" : 100,
  "counts" : {
        "input" : 2,
        "emit" : 2,
        "reduce" : 1,
        "output" : 1
  },
  "ok" : 1
```

EMERTXE

Single purpose aggregation operations

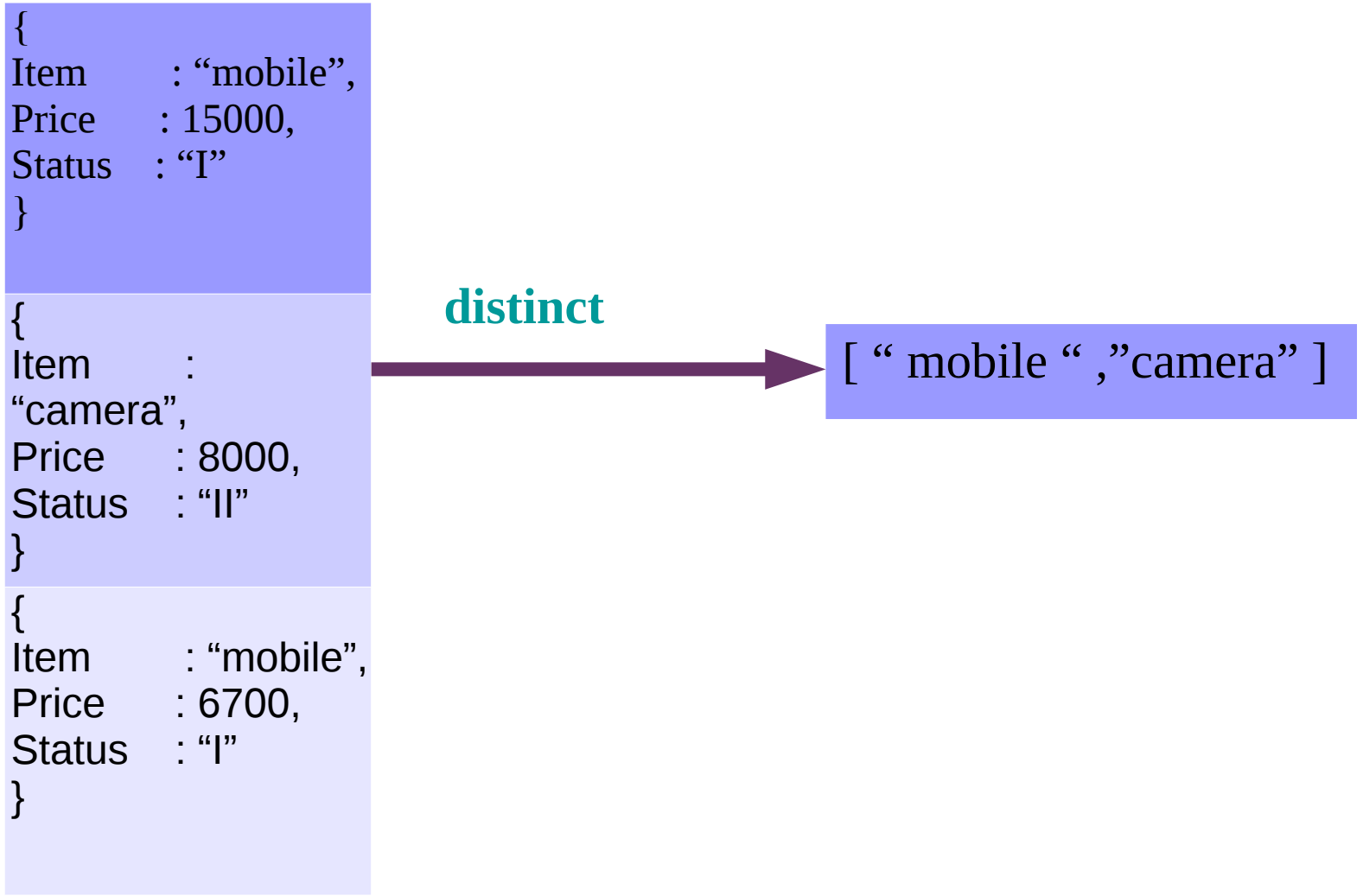
These operations aggregate documents from a single collection.

These operation lack the capabilities of aggregation pipeline and map-Reduce.

Example : db.collection.count() ,

db.collection.distinct()

Single purpose aggregation operations



SQL term to Aggregation mapping chart

SQL Terms	MongoDB Aggregation Operator
WHERE	$match
GROUP BY	$group
HAVING	$match
SELECT	$project
ORDER BY	$sort
LIMIT	$limit
join	$lookup
SUM	$sum

References

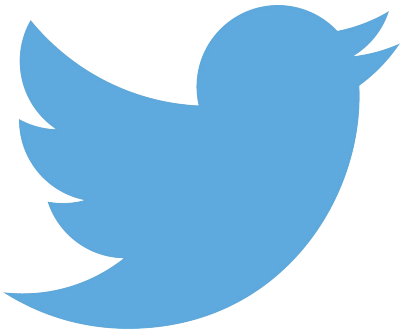
- https://www.wikimedia.org/
- https://docs.mongodb.com/manual/

Stay connected

About us: Emertxe is India's one of the top IT finishing schools & self learning kits provider. Our primary focus is on Embedded with diversification focus on Java, Oracle and Android areas

Emertxe Information Technologies,
No-1, 9th Cross, 5th Main,
Jayamahal Extension,
Bangalore, Karnataka 560046
T: +91 80 6562 9666
E: training@emertxe.com

https://www.facebook.com/Emertxe



https://twitter.com/EmertxeTweet

https://www.slideshare.net/EmertxeSlides

EMERTXE

Thank You