db.purchase\_orders.insertMany([

{product: "toothbrush", total: 4.75, customer: "Mike"},

{product: "guitar", total: 199.99, customer: "Tome"},

{product: "milk", total: 11.33, customer: "Mike"},

{product: "pizza", total: 8.50, customer: "Karen"},

{product: "toothbrush", total: 4.75, customer: "Karen"},

{product: "pizza", total: 4.75, customer: "Dave"},

{product: "toothbrush", total: 4.75, customer: "Mike"}

])

find out how many toothbrush were sold

db.purchase\_orders.count({product:"toothbrush"})

find list of all products sold

db.purchase\_orders.distinct("product")

find the total amount of money spent by each customer

db.purchase\_orders.aggregate(

[

{$match:{}},

{$group: {\_id: "$customer", total: {$sum: "$total" }}},

{$sort: {total: -1}}

]

)

Aggregation operations process data records and return computed results. Aggregation operations group values from multiple documents together, and can perform a variety of operations on the grouped data to return a single result.

MongoDB provides three ways to perform aggregation:

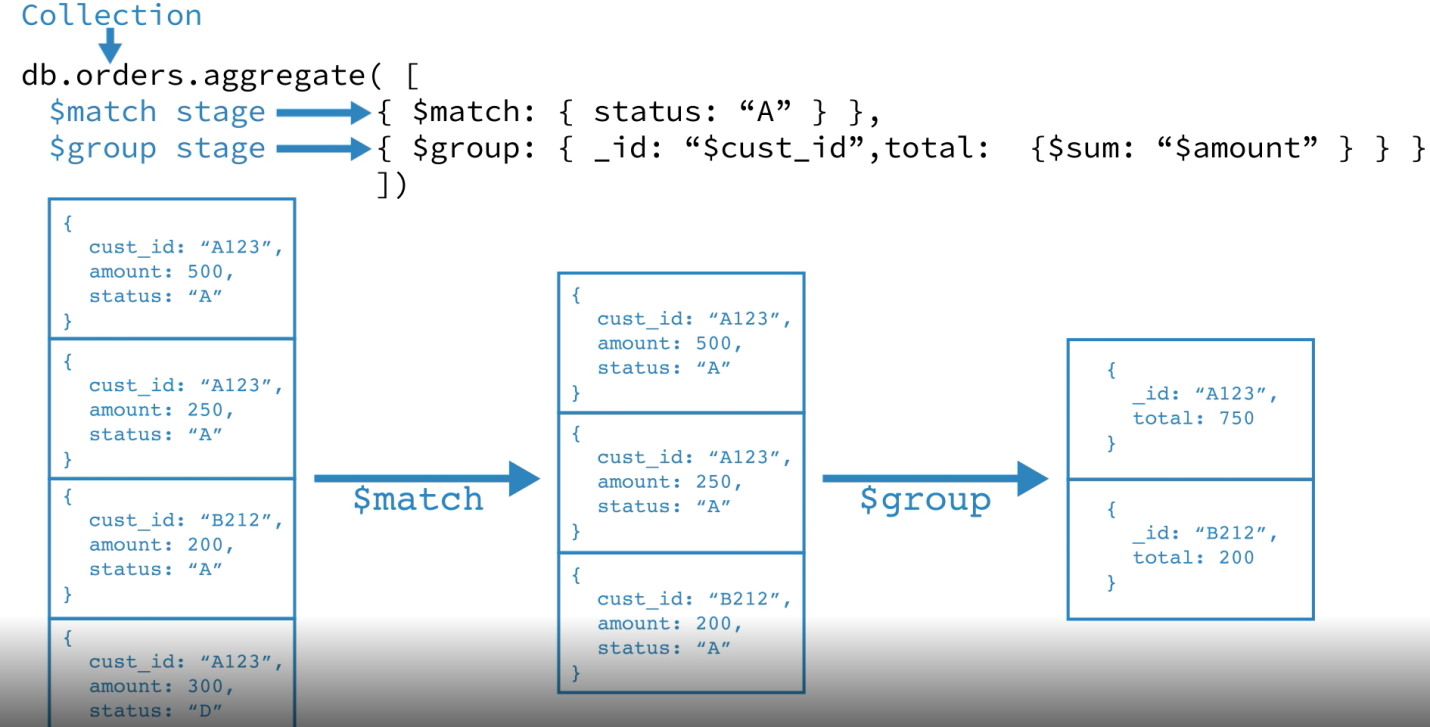
* [**Aggregation pipeline**](https://docs.mongodb.com/manual/aggregation/#aggregation-framework)**:-** MongoDB’s [aggregation framework](https://docs.mongodb.com/manual/core/aggregation-pipeline/) is modeled on the concept of data processing pipelines. Documents enter a multi-stage pipeline that transforms the documents into an aggregated result.

db.orders.aggregate([

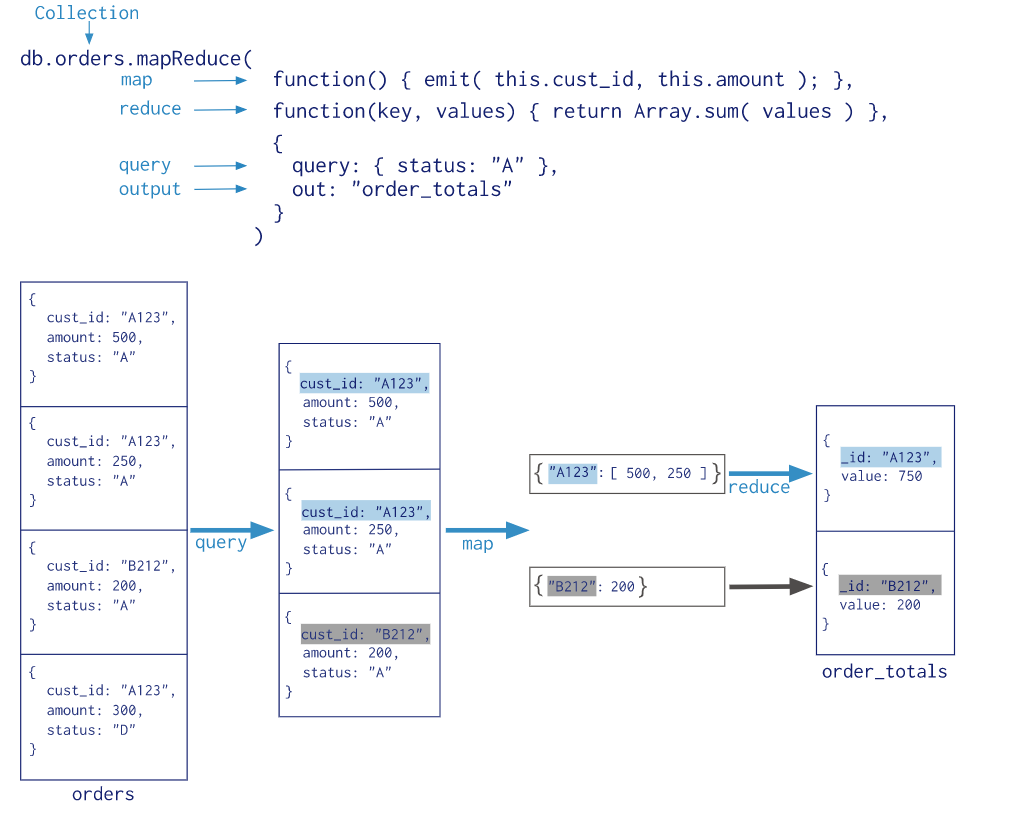
{ $match: { status: "A" } },

{ $group: { \_id: "$cust\_id", total: { $sum: "$amount" } } }

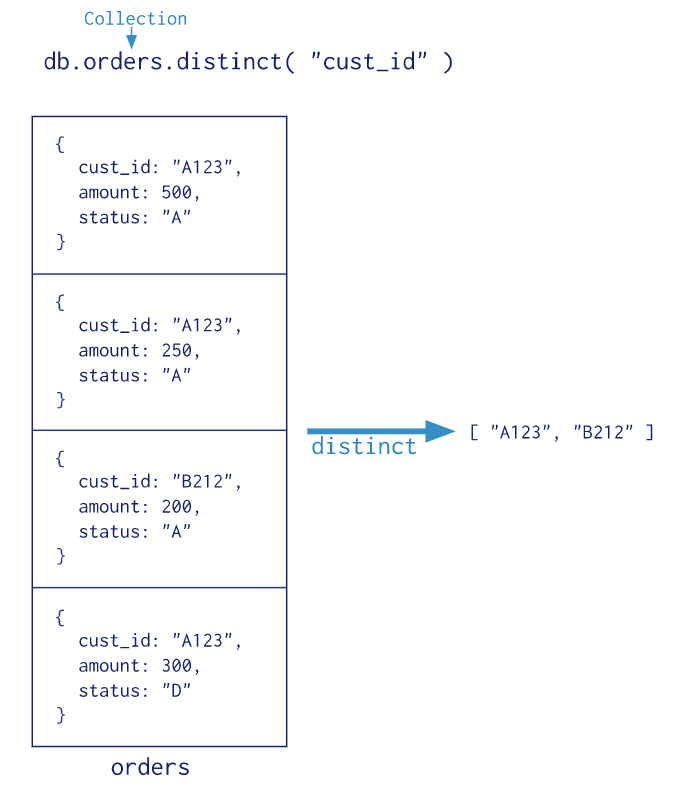
])



[**map-reduce function**](https://docs.mongodb.com/manual/aggregation/#aggregation-map-reduce)**:-**  Map-reduce uses custom JavaScript functions to perform the map and reduce operations, as well as the optional *finalize* operation.



* [**single purpose aggregation methods**](https://docs.mongodb.com/manual/aggregation/#single-purpose-agg-operations)**.** : - While these operations provide simple access to common aggregation processes, they lack the flexibility and capabilities of the aggregation pipeline and map-reduce.



find the total amount of money spent by Mike and Karen

db.purchase\_orders.aggregate( [

{$match:{customer: {$in: ["Mike", "Karen"]}}},

{$group: {\_id: "$customer", total: {$sum: "$total" }}},

{$sort: {total: -1}}

])