Aggregation → data processing Pipeline

```
db.collection.aggregate([
        {Stage 1},
        {Stage 2},
         {Stage 3}
    ])
```

different Stages

Stage 1 $\xrightarrow[\text{then}]{}$ Stage 2 → ...

Stage	Purpose	SQL Like
$match	Filter	Where
$project	Compute	Select
$group	Group + Aggregate	Group By
$Sort	Sort	Order By
$Limit	limit	Limit
$lookup	Join	Join

Eg:
```
db.orders.aggregate([
  {$group: {
        _id: "$customer",
        totalSpent: {$sum: "$amount"},
        Ordercount: {$sum: 1}
  }}])
```

Select
Sum(AMT)
as total

```
{$match: {status: "completed"}},
 {$sort: {totalSpent: -1}}
```

```
db.orders.aggregate([{
      $lookup: {
            from: "customers",            ] JOIN customer
            localField: "customerId",     ]  on a = b
            foreignField: "_id",          ]
            as: "customer"
      }}])
                     ⟶  join as field
                          customer
```