

MongoDB Aggregate Quick Revision Notes

MongoDB aggregate() - Quick Revision

Basic Syntax:

```
db.collection.aggregate([ { stage1 }, { stage2 }, ... ])
```

Important Stages:

\$match -> Filter documents.

```
{ $match: { age: { $gte: 18 } } }
```

\$group -> Group by field and use accumulators.

```
{ $group: { _id: "$category", total: { $sum: 1 } } }
```

\$project -> Select or reshape fields.

```
{ $project: { name: 1, year: { $year: "$date" } } }
```

\$sort -> Sort documents.

```
{ $sort: { age: -1 } }
```

\$limit -> Limit results.

```
{ $limit: 5 }
```

\$skip -> Skip documents.

```
{ $skip: 10 }
```

\$count -> Count number of documents.

```
{ $count: "total" }
```

\$lookup -> Join collections.

```
{  
  $lookup: {  
    from: "otherCollection",  
    localField: "userId",  
    foreignField: "_id",  
    as: "userData"  
  }  
}
```

\$unwind -> Break arrays into individual documents.

```
{ $unwind: "$items" }
```

MongoDB Aggregate Quick Revision Notes

\$addField -> Add new/computed fields.

```
{ $addField: { total: { $multiply: ["$price", "$qty"] } } }
```

Example Pipeline:

```
db.sales.aggregate([  
  { $match: { status: "active" } },  
  { $group: { _id: "$category", totalSales: { $sum: "$amount" } } },  
  { $sort: { totalSales: -1 } },  
  { $project: { category: "$_id", totalSales: 1, _id: 0 } }  
])
```

Accumulator Functions:

\$sum, \$avg, \$min, \$max, \$push, \$addToSet, \$first, \$last

Tips:

- Use \$match early for performance.
- Aggregation is powerful for analytics and reporting.